ANNO 1 - MENSILE - LIRE 4.000

AMIGA Readme

TUTORIAL

N. 1 - OTT / NOV '89



SMAU '89

Workbench 1.4

Page Stream

Design 3D

Krystal

Dungeon Master

VGM

Amiga da zero

I compattatori

SMAU 89

Appena in tempo, abbiamo deciso di inserire le nostre impressioni su quello che può definirsi senz'altro l'appuntamento nazionale più importante dell'anno per gli operatori e gli appassionati di informatica. La mia prima sensazione è stata la conferma del grande interesse del pubblico giovane, ma non solo, per l'Amiga e le sue possibili applicazioni. Novità vere e proprie non ne ho viste molte, tranne il **PAL-RGB CONVERTER** della P.G. Elettronica Torino e il **P.F.M.** della Cloanto di Udine. Tuttavia, non sono scontento o deluso da questo salone, come credo che non lo siano la maggior parte degli utenti Amiga che lo hanno visitato.



La nota positiva è stata la intraprendenza di ditte che, non legate in modo diretto alla Commodore, hanno dimostrato una notevole vitalità, immettendo sul mercato italiano una serie di prodotti già presenti negli USA.

Particolarmente vantaggiose le offerte della HARDITAL: il **Flicker Fixer**, che consente di usare qualunque programma in alta risoluzione senza il fastidiosissimo sfarfallio, del quale speriamo di potervi fornire una prova completa al più presto; il **Modulo Autoboot** per il controller hard disk A2090; l'hard disk per Amiga 2000 **Flashbank** da 32 MB 40 Ms con autoboot venduto al pubblico al prezzo di lire 690.000 iva compresa e l'espansione di memoria **Superotto**, sempre per A 2000, con 2 megabyte, zero wait state, autoconfigurante, espandibili ad otto al prezzo di lire 740.000 iva compresa. Grande interesse ha destato la dimostrazione dell'emulatore Macintosh A-MAX distribuito dalla SPY di Napoli di cui abbiamo parlato il mese scorso e che proveremo per voi: in breve è possibile con una spesa che si aggira sulle settecentomila lire avere un vero e proprio Machintosh Plus con tanto di drive compatibile.



Attiva, come sempre, la Newel di Milano ha presentato una vasta gamma di prodotti importati dagli USA ed in particolare la scheda A3001 con 68030 prodotta dalla G.V.P.; una scheda simile Commodore equipaggerà i nuovi Amiga 2500UX.

Davvero simpatica è stata l'iniziativa degli amici di Enigma, che, sfruttando anche loro un sistema di DTP basato su Amiga 2000, hanno impaginato e stampato in pubblico un bollettino di aggiornamento sulle novità Commodore allo SMAU.



Per quanto riguarda il software, settore nel quale i problemi sono davvero tanti, ci lascia intravedere un futuro migliore l'impegno con cui la CTO si sta dedicando alla commercializzazione di prodotti di notevole valore, quali ad esempio quelli della Gold Disk. Presenti alla manifestazione anche la LEADER, la Simulmondo e la Cloanto. Grande assente, ma speriamo che non lo sia almeno sul mercato, la Lago.

Continua in III di copertina

Readme First!

BENVENUTI nell'anno dell'AMIGA

Nel presentare questo numero uno della nostra rivista desidero innanzitutto ringraziare quanti, nonostante le imperfezioni tipografiche e redazionali del numero zero, hanno voluto accordarci la loro fiducia, premiando il nostro impegno e la qualitá dei contenuti.

Ci é sembrato doveroso compiere un ulteriore sforzo, economico ed organizzativo, per realizzare un prodotto sempre migliore. Siamo giá al lavoro ancora e possiamo assicurarvi che di mese in mese faremo il possibile per rendere README sempre piú utile e ricca di contenuti.

Questo inverno sará una stagione davvero importante per Amiga.

Le numerose novitá hardware e software (in primis, comunque, i sospirati ECS e Workbench/Kickstart 1.4) dovrebbero finalmente scrollare di dosso all'Amiga la fama di sofisticato "giocattolone".

Perché questo avvenga dovremo, noi tutti utenti Amiga, risolvere un problema di fondo: scoprire il motivo che ci spinge ad utilizzare il computer e decidere l'uso che ne vogliamo fare.

Conosciamo una gran quantitá di persone che collezionano programmi. Accumulano centinaia e centinaia di dischetti pirata, spesso non funzionanti e che, dopo una occhiata frettolosa, non useranno mai. Si possono spendere cifre anche ragguardevoli per comprare tale spazzatura. Con duecentomila lire al mese, il costo medio di un abbonamento al mercato nero del software, é possibile acquistare libri, dischi e qualunque altro oggetto che dia soddisfazioni meno effimere del veder girare nel proprio drive la versione 78.09 del famoso compilatore C che non useremo mai. Senza considerare poi quella che sarebbe la scelta piú intelligente: spendere i duemilioniquattrocentomila lire annue per l'acquisto di quel software originale che realmente useremo e, credeteci, con quella cifra (di poco superiore al costo del solo Pagemaker IBM o Mac) avremmo il meglio del software Amiga in circolazione.

Certo, ci sarebbe l'inconveniente di risparmiare un bel po' di soldi, ma si potrebbe ovviare al "problema" acquistando nuovo hardware o aprendo un bel conto corrente.

Senza dimenticare la soddisfazione che si prova ad utilizzare un computer sofisticato come l'Amiga con programmi di cui, grazie ai manuali, si conosca ogni funzione e si sappia come ottenere il risultato migliore,.

E' di poco tempo fá la notizia secondo cui la Worperfect Corporation (autrice del word processor piú venduto al mondo per MS/DOS) annunciava di voler chiudere il reparto progettazione e sviluppo del Word Perfect per Amiga, scoraggiata dalle sue scarse vendite. Solo dopo una vera e propria valanga di lettere e proteste degli utenti USA, che evidentemente usano l'Amiga non solo per giocare, la software house americana ha modificato la sua decisione ed ha annunciato la prossima uscita di Word Perfect 6.0 e dello spreadsheet Plan Perfect.

Riflettiamo su ció se non vogliamo diventare, tra qualche anno, i possessori di quello che sarebbe potuto essere il piú sofisticato personal computer esistente.

Readme.doc

Potete scriverci al seguente indirizzo:

CompWare
via N. Piccinni, 2
80129 Napoli

oppure telefonare al numero:

081-578.64.99

News

La Gold Disk ha sviluppato un'interfaccia hardware e software per lo scanner in bianco e nero CANON IX12. Il software permette di salvare le immagini "scannerizzate" in formato IFF a 300, 200, 150 e 75 dpi. Lo scanner e l'interfaccia costano 1095 dollari usa.
GOLD DISK P.O.BOX 789, Streetville, Mississauga, ONTARIO, CANADA

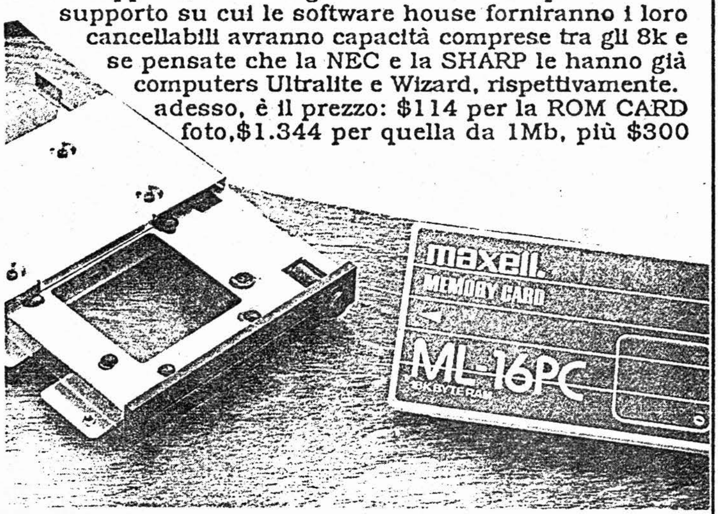
3D OPTIONS è un nuovo programma che converte immagini IFF bitmap in oggetti bidimensionali o tridimensionali. Tramite esso è possibile portare qualsiasi immagine, in un programma di grafica 3D o in un programma di Cad, tipo Videoscape, Aegis Draw, Introcad, Professional Page, ecc. Il programma costa circa 50 dollari usa.
RAINBOW EDGE PRODUCTIONS 4412 4th Avenue, Suite 2 Brooklyn, NY 11220

ELAN PERFORMER è un programma rivolto alle video presentation, che ci permette di realizzare sequenze animate pilotate da tastiera. Lo scambio tra uno schermo e l'altro non avviene tramite complessi files script o tramite menu, ma semplicemente premendo un tasto. Elan Performer utilizza un semplice schermo di controllo per assegnare ciascuna immagine od un'intera animazione ad un tasto . Se vogliamo, possiamo automatizzare completamente la sequenza animata, oppure usare il mouse per muoverci avanti e indietro attraverso le varie immagini che compongono il nostro slideshow. Il programma ci permette inoltre di scompattare le animazioni nei singoli frames che le compongono, e converte animazioni dal formato Riff in Anim e viceversa. Elan Performer supporta i formati grafici IFF, HAM, RIFF, ANIM e RGB. Lavora anche con soli 512k. Supporta l'overscan per le applicazioni videografiche. Il prezzo di Elan Performer è di 59 dollari usa.
ELAN DESIGN P.O.BOX 31725 San Francisco, CA 94131

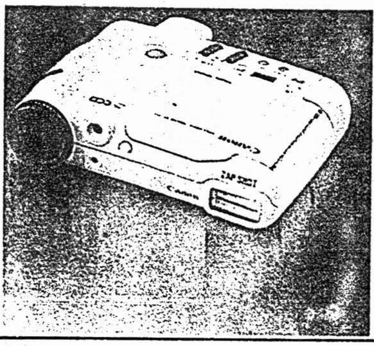
La Newtek ha lanciato sul mercato statunitense l'ultima versione del programma di grafica pittorica DIGI-PAINT 3. Tre perchè, come spiega Tim Jenison, presidente della Newtek, questa nuova versione è tre volte migliore della precedente e perchè è costata tre anni di lavoro. In effetti il programma è stato profondamente migliorato in ogni sua funzione. La realizzazione dell'interfaccia utente dello schermo di lavoro ha richiesto da sola mesi di lavoro, ed è stata implementata da Jim Sachs, uno dei migliori Amiga's artist del mondo. In Digi-paint 3 è stato implementato un modo di disegno SUPER BIT MAPS, grazie al quale possiamo creare immagini di 1024x1024 pixels. Ovviamente sullo schermo, disegnando in questo modo grafico, vedremo solo una parte del disegno. Digi-paint 3 lavora in modo HAM, con una palette di 4096 colori visualizzabili contemporaneamente. E' inclusa nel pacchetto l'utility TRANSFER 24, che trasforma l'Amiga in una workstation per manipolare immagini. Transfer 24 converte i disegni Ham in ogni risoluzione e modo grafico (incluso l'halfbrite e l'hi-res); essa permette inoltre un completo controllo dei livelli RGB, del contrasto, della luminosità, saturazione, e l'abilità di rimappare le immagini in base a qualsiasi palette. In Digi-paint 3 possiamo inserire testi e manipolare qualsiasi tipo di font anche a colori. Digi-paint 3 è stato scritto interamente in linguaggio Assembler, per renderlo più veloce. E' stato predisposto anche per lavorare sotto ambiente Arexx. Digi-paint 3 costa 100 dollari usa.
NEWTEK 115 W.CRANE ST. TOPEKA, KANKAS 66603 - USA-

uno sguardo al futuro...

Si avvicina forse la fine del floppy disk. Guardate la foto: quella che vedete non è una nuova carta di credito, anche se le dimensioni sono le stesse. Collegate il drive, cioè quell'oggetto che vedete sulla sinistra, alla porta parallela del vostro personal ed inseritevi la carta: ebbene avrete un supporto circa 40.000 volte più veloce del floppy disk. Inoltre, non essendoci nel drive parti in movimento, si riducono quasi a zero i rischi di errori di lettura scrittura o di danneggiamento del supporto. Le ROM CARDS, questo il nome dei nuovi supporti, esistono già in due versioni: quelle a sola lettura costituiranno il supporto su cui le software house forniranno i loro programmi; quelle cancellabili avranno capacità comprese tra gli 8k e 1Mb. Fantascienza? No, se pensate che la NEC e la SHARP le hanno già utilizzate nei pocket computers Ultralite e Wizard, rispettivamente. Unico ostacolo, per adesso, è il prezzo: $114 per la ROM CARD da 128k che vedete nella foto,$1.344 per quella da 1Mb, più $300 per il lettore.



Tra poco arriverà anche in Italia la nuova fotocamera compatta CANON Zapshot, che utilizza floppy disks da 2" al posto della pellicola fotografica. Su questi dischetti la Zapshot è in grado di memorizzare 50 immagini a colori. Il vantaggio, oltre al fatto che è possibile sostituire il disco alla luce del sole, è che si possono rivedere immediatamente le immagini semplicemente collegando la Zapshot ad un comune televisore o riversandole su una videocassetta.



Immaginate adesso un collegamento Zapshot-Amiga tramite un digitalizzatore: via filtri e telecamere: basterà caricare le immagini appena scattate con uno dei numerosi programmi di grafica per trasformarle a vostro piacimento!

Il nome completo di questo piccolo mostro è CANON RC-250 High Band Still Video Camera.

Readme

ANNO I - Numero 1
Ottobre 1989

Pubblicazione periodica mensile

Direttore Responsabile
Carlo Tripodi

Direttore Editoriale
Ettore Lapegna

Grafica ed Impaginazione
Maurizio Avati

Redattore Capo
Massimiliano Afiero

Hanno Collaborato
Pasquale Alfuso
Fabrizio Del Gaudio
Lelio Della Pietra
Luca Di Meo
Maria La Marca
Luigi Lanza

CompWare Edizioni
via N. Piccinni, 2 NAPOLI
tel. 081/5786499

Rivista in corso di registrazione presso il Tribunale di Napoli

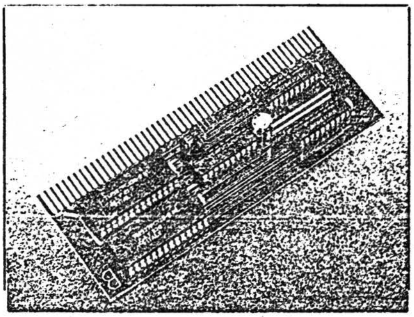
Stampa
Pesole S.N.C.
via Mezzocannone, 93
Napoli

Distributore per l'Italia
ADP s.r.l.
Via Leopoldo Traversi, 39
Roma

Sommario

Commodore

A2090B



Grosse novità annunciate dalla Commodore durante l'ultima Developer's Conference, tenutasi a San Francisco dal 14 al 18 giugno scorso:

- scheda A2630 per Amiga 2000 che vedete, non molto chiaramente, in verità, nella foto in basso a destra tra le mani del "papà" Dave Haynie. Le caratteristiche: processore Motorola 68030, coprocessore matematico 68882; clock a 25MHz; 2Mb di RAM a 32 bit espandibile a 4Mb sulla scheda.

- Scheda grafica A2360, progettata da Rick Miner della Lowell University in collaborazione con la Commodore, con risoluzione di 1024x768 punti e 256 colori da una palette di 16,8 milioni, basata sul chip TI 34010 della Texas Instruments.

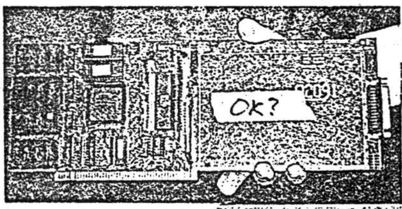
- scheda multiseriale A2060 da utilizzare con il sistema X-Windows

- kit A2090B, che fornisce l'autoboot ai vecchi controller per hard disk A2090 (nella foto in alto).

- hard-card A2091 che monta un hard disk direttamente sulla scheda del controller, senza occupare quindi lo spazio riservato al secondo drive da 3.5".

Si comincia a delineare, intanto, la fisionomia dell'Amiga 3000: processore 68030, bus dati a 32 bit, invece dei 16 dei modelli attuali, 1Mb Chip RAM.

Ma tutti aspettiamo ansiosi la versione 1.4 del sistema operativo e l'ECS (Enhanced Chip Set) che apporteranno modifiche sostanziali al nostro Amiga: date uno sguardo alla pagina seguente!





Dave Haynie e la scheda A2630

Hard card A2091

News

Workbench 1.4

Grande attesa fra tutti gli utenti Amiga per l'ECS e per il Workbench 1.4.

La Commodore, in occasione dell'ultima Developers' Conference a San Francisco, ha svelato quale saranno le modifiche e le aggiunte al sistema operativo, che, stando alle parole di Andy Finkel, sarà quasi completamente riscritto.

La parola d'ordine di questo progetto è "compatibilità": in pratica tutto il software funzionante sotto 1.3 sarà compatibile anche con l'1.4. Non funzioneranno solo quei programmi che, contravvenendo alle direttive Commodore, accedono a strutture dati privilegiate del sistema operativo.

La differenza più evidente sarà la nuova interfaccia grafica: il nuovo presidente della Commodore Harry Copperman, proveniente dalla Apple e con un passato in IBM, ha infatti dichiarato di aver ingaggiato un "artista" per dare al Workbench un aspetto più professionale. Inoltre lo schermo Workbench avrà una sua finestra dimensionabile il cui sfondo sarà costituito da un retino, ridisegnabile dall'utente (come sui Mac e sull'Atari ST). Sulla barra dei menu compaiono le indicazioni della quantità di memoria chip, della memoria fast e dell'ora di sistema.
Sarà poi possibile visualizzare da Workbench anche quei files sprovvisti di icona grazie ad un'apposita opzione presente in un nuovo menu VIEW: altre voci dello stesso menu consentiranno l'ordinamento di questi files in base al nome, alle dimensioni e alla data di creazione. Ma non basta: se il disco contiene delle icone e si è selezionata la visualizzazione testo, sarà comunque possibile far partire quei programmi semplicemente cliccando sul nome!
Un secondo nuovo menu contiene l'unica opzione RESET WB che consente di riaprire lo schermo Workbench con nuovi parametri (risoluzione, font di sistema, colori) senza dover resettare l'intero sistema.
La directory WBS contiene i tool che permettono di comunicare all'Amiga la presenza del monitor Commodore A2024 o di un monitor Multisync.

Nuovo anche il look delle finestre: è scomparsa la barra verticale che indicava il grado di "riempimento" del disco, sostituita da indicazioni numeriche molto più complete ed utili: % di riempimento, spazio libero in Kylobytes e spazio totale. A sinistra dei gadgets di profondità compare un nuovo gadgets che serve a visualizzare nella finestra il contenuto della directory di livello superiore: comodissimo.

Migliorerà anche il grado di multitasking del Workbench, nel senso che sarà possibile far partire più programmi contempo- raneamente da Workbench con la tecnica del doppio click sull'icona.

Altra attesa novità sarà l'estensione del Fastfilesystem (FFS) ai floppy disk grazie alla sua inclusione nel Kickstart. Il vecchio file system sarà comunque presente per mantenere la compatibilità.

Il nuovo sistema operativo supporterà l'Enhanced Chip Set costituito principalmente dal Super Agnus e dal Super Denise; essi rimpiazzeranno i corrispondenti chips dei 500 e dei 2000 (la sostituzione pare sarà impossibile con l'Amiga 1000). Attualmente la Commodore sta già fornendo i 500 e i 2000 equipaggiato con il nuovo Agnus, in grado di indirizzare 1 Mb di memoria grafica (Chip Ram).

Il nuovo Denise consentirà i nuovi modi grafici, compreso il productivity mode con la risoluzione 640x480 non interlacciata, a 4 colori scelti da una palette di 64; per questo modo è comunque necessario il monitor A2024 o un multisync. Gli altri modi prevedono la risoluzione 1280x200/256 (1280x400/512 interlacciati), sempre con 4 colori su 64. Il nuovo Denise migliorerà anche la gestione dei genlock, con la possibilità di genlockare ogni singolo bitplane e di poter settare ogni registro colore come trasparente.

Verrà aumentata la dimensione del blit a 32x32 Kb (attualmente è di 1x1 Kb) che consentirà animazioni più fluide e veloci.

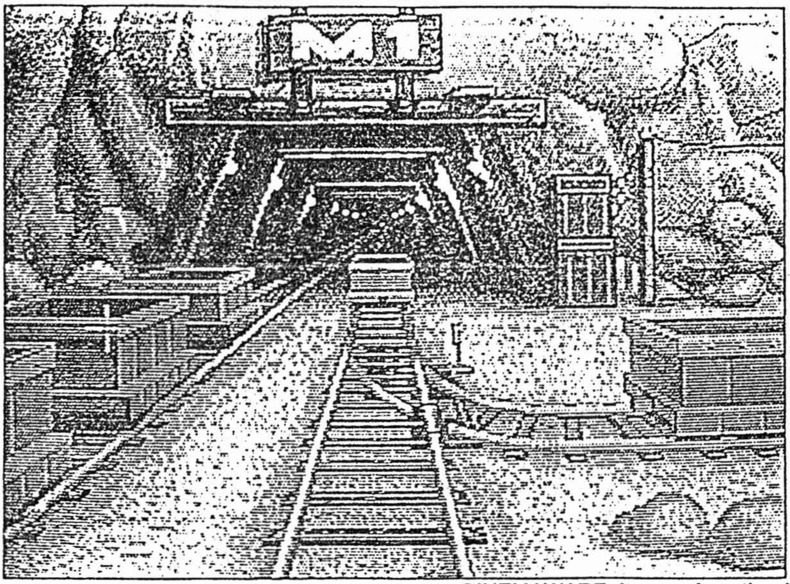
L'AREXX, il linguaggio di comunicazione tra processi importato dal mondo IBM da William Hawes, sarà già implementato sul Workbench. Grazie a questa importante feature sarà, ad esempio, possibile importare in tempo reale in uno spreadsheet informazioni provenienti da un programma di comunicazione e riordinarle in un wordprocessor, purché questi programmi siano AREXX-compatibili.

La Commodore, inoltre, incoraggerà gli sviluppatori ad utilizzare i requester di sistema anziché quelli personali, anche se non raggiungerà mai la rigidità della Apple; questo perché l'aver ridisegnato il Workbench e lo sviluppo dei files e dei fonts requester standard dimostrano che la Commodore ha capito l'importanza di fornire agli utenti un'interfaccia più "consistente".

Quando usciranno ECS e Workbench 1.4?

La Commodore non si sbilancia nel precisare date esatte. Nostre fonti, abbastanza attendibili, dicono non prima del CEBIT '90, in marzo.

GamesNews



CINEMAWARE: It came from the desert

Sono attesi a momenti **XENON II** e **Megablast**, rispettivamente i seguiti di Xenon e Speedball. Il primo conterrà sei livelli zeppi di alieni e conserverà lo scrolling verticale della prima versione.

Dalla Mirrorsoft, invece, aspettiamo **Bloodwych**, una sorta di versione per due giocatori di Dungeon Master: lo schermo di gioco è diviso esattamente a metà e ciascuna porzione mostra l'ambiente dal punto di vista di ciascun giocatore. Ovviamente ci sono miriadi di personaggi da incontrare, pagine e pagine di formule magiche da pronunciare al momento opportuno.

La prossima fatica della Cinemaware ci porterà indietro nel tempo e precisamente negli anni '50 in una sorta di horror-movie. Il titolo è tutto un programma: "**It came from the desert**" (Arrivò dal deserto"). La trama: subito dopo la caduta di una meteora nel deserto dell'Arizona, cominciano ad accadere strane cose. Naturalmente c'è il professore che fà previsioni catastrofiche cui nessuno crede, almeno fino a quando un'enorme formica non inghiotte l'intera città. Il giocatore impersona un giovane scienziato curioso deciso a sterminare gli odiosi insetti. I programmatori, i disegnatori ed i musicisti alla Cinemaware sembrano decisi a dare il meglio di se stessi: a giudicare dalla schermata giuntaci con un demo del gioco, almeno i disegnatori ci sono riusciti!
Dalla PSS arriva invece **WATERLOO**, simulazione della famosa battaglia con il più realistico 3D mai visto in un war-game.

Dalla Spectrum Holobyte due novità: il primo disco scenario per l'ottimo simulatore di volo FALCON e **VETTE** che utilizza le avanzate tecniche di grafica vettoriale di Falcon per una simulazione di guida. Il giocatore è alla guida di una Chevrolet Corvette nelle strade di San Francisco, con la possibilità di scegliere la veduta dell'auto: aerea, da un

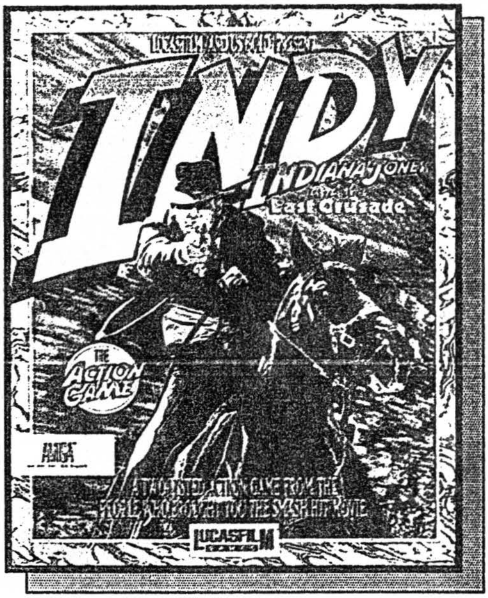
elicottero, laterale e frontale. Probabilmente nei negozi da quest'inverno.

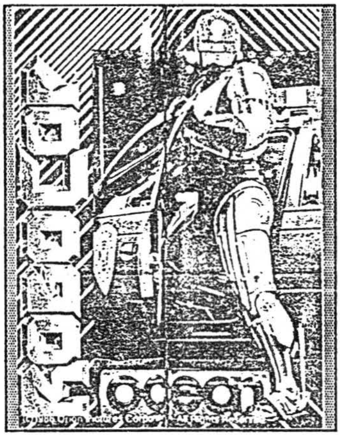
Ritorna Garfield in nuova avventura: WINTER'S TAIL.

Prima ancora del film, interpretato come al solito da Harrison Ford, affiancato questa volta dal grande Sean Connery, arriva sugli schermi Amiga INDIANA JONES AND THE LAST CRUSADE della LucasFilm Games.

Dalla Core Design, distribuita dalla Firebird, un ennesimo arcade. RICK DANGEROUS il titolo.

Molto atteso dopo il successo del film omonimo, arriva ROBOCOP, in versioni per Amiga, Atari ST, IBM, Spectrum ecc., tanto per non scontentare nessuno.







News

Scanlock



La VidTech International ha presentato un nuovo genlock di qualità broadcast per Amiga.
Lo Scanlock VSL-1, questo il suo nome, confronta il segnale video di Amiga con quello di un sistema di riferimento (in NTSC o in PAL) e poi miscela il segnale di Amiga con quello di riferimento.
Lo Scanlock utilizza le più aggiornate tecniche analogiche e digitali che offrono finalmente anche agli utenti Amiga una qualità d'immagine degna dei sistemi professionali.
Il segnale che Scanlock utilizza come riferimento può essere quello proveniente da una videocamera o da un videorecorder. Inoltre è in grado di processare sia segnali compositi (NTSC o PAL) che i componenti (luma/croma o Y/C), incluso il nuovo S-VHS, M2 o BETACAM, trasformando il segnale RGB di Amiga senza interferire con la qualità del segnale di riferimento.

Scanlock VSL-1 è disponibile nei due formati, NTSC e PAL, contrassegnati rispettivamente dalle sigle VSL-1-N e VSL-1-P.

Questo sofisticato genlock trasforma la grafica del computer in modo da sincronizzarla con quella dell'altra sorgente video. Inoltre Scanlock offre un amplificatore per il segnale del computer con possibilità di calibrazione sufficientemente ampia. Esiste la possibilità di controllare le funzioni del genlock da un secondo computer o da un controllore di editing.

Scanlock ha il controllo del FADE, indipendente per i segnali video del computer e dell'altra sorgente video, direttamente sul pannello frontale; offre la scelta tra l'alimentazione indipendente o tramite l'alimentatore di Amiga tramite uno switch presente sul pannello frontale.

Broadcast Titler

Siamo tutti in attesa di vedere all'opera l'ultima fatica dell'Innovision (quella di VIDEO EFFECTS 3D), che si preannuncia davvero portentosa: risoluzione 2160x1440, accetta i fonts standard Amiga e i Colorfonts, trasformandoli in alta definizione fino alla massima risoluzione; 320 colori da una palette di 4096 in alta risoluzione; piu' immagini IFF sulla stessa pagina; 16 colori per ogni linea di testo; sfondi e patterns a colori automatici; 100 modi di transizione della riga e della pagina; fino a 1.000 pagine di grafica e di testo. Tutto cio' per soli $299.95.

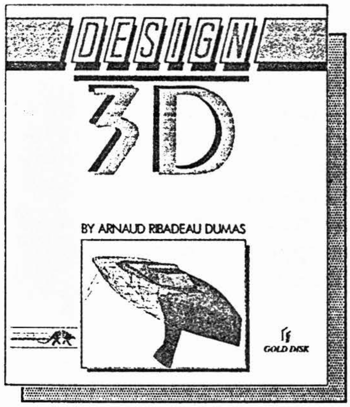
Innovision Technology - P.O. Box 743 - Hayward, CA 94543

Design 3D

Esistono numerosi programmi di 3D-modeling per Amiga, ma, secondo noi, solo l'Aegis Modeler ha il pregio di essere più intuitivo da utilizzare rispetto ai tools "di serie" in programmi tipo Sculpt-Animate 4D e Videoscape 3D.

Oggi il suo primato viene messo in serio pericolo da Design 3D. La canadese Gold Disk si sta sempre più affermando come la software house che, più di ogni altra, si sforza di fornire ad Amiga programmi degni delle sue capacità. Prima con Professional Page, poi con Professional Draw (di cui attendiamo una versione migliore, a livello dei vari Illustrator '88 e Freehand per Mac), ancora con Transcript, infine con Design 3D, l'oggetto di questa veloce recensione.

Tutte le funzioni principali del programma e gli strumenti sono accessibili per mezzo di icone disposte su tre dei lati dello schermo di lavoro. Quest'ultimo è diviso in quattro porzioni, una per ogni veduta dell'oggetto, dall'alto, frontale, laterale e in prospettiva; ogni porzione può essere ingrandita a tutto schermo.
Gli strumenti e le funzioni comprendono tutto ciò che ci si aspetta da un programma del genere ed anche di più: disegno delle figure tridimensionali fondamentali (poligoni, rettangoli, ellissi, archi, ecc.), cloning, che consente di ruotare un oggetto attorno ad un asse verticale, il point merging che consente di riunire più punti dell'oggetto, l'estrusione, rotazione e, infine, gli strumenti per il movimento, la scalatura, la cancellazione e la copia dei punti, dei poligoni e degli oggetti.



Design 3D dispone inoltre di numerosi tools comuni nei programmi di disegno 2D, quali il fill e il disegno a mano libera, che consentono di ritoccare l'immagine 3D. Molto interessante è poi la possibilità di creare testo direttamente in 3 dimensioni, forse la caratteristica migliore del programma: basta selezionare l'icona testo, inserire il testo nel requester

apparso sullo schermo di lavoro e poi tracciare su quest'ultimo i box in cui il testo verrà adattato. Inoltre Design 3D dispone di un editor di fonts 3D.

I menu contengono, invece, le selezioni di minor frequente impiego: il tipo di risoluzione, la scelta dei 16 colori da una palette di 4096, i parametri che il programma leggerà alla partenza, tra cui l'unità di misura (pollici, centimetri, chilometri, ecc.), ledimensioni della griglia e la posizione delle sorgenti di luce. Altre opzioni da menu comprendono il rapporto di riproduzione in stampa, la scelta della stampante o del plotter, la scelta tra la visualizzazione dell'oggetto esclusivamente in wire-frame o con i colori e le gradazioni di colore e le ombre scelte.

Oltretutto Design 3D permette l'animazione degli oggetti, anche se non nel modo sofisticato dei pacchetti del livello di Sculpt o Videoscape. Esiste comunque la possibilità di esportare gli oggetti creati con Design 3D in Videoscape e viceversa.

Un'ultima osservazione: oltre al manuale, esauriente come è costume della Gold Disk, su disco troverete un utilissimo tutorial animato che, oltre a servire da testimonianza delle capacità del programma, consente anche all'utente più sprovveduto di iniziare a creare semplici oggetti tridimensionali senza leggere il manuale, cosa che comunque consigliamo di fare non appena avrete acquisito la sufficiente dimestichezza con Design 3D.

Design 3D
Gold Disk, Inc.
P. O. Box 789
Streetsville, Mississauga
Ontario, Canada.

$ 99.95
Richiede 1 Mb

I compattatori

TEORIA E METODI DI COMPATTAZIONE DEI FILES

INTRODUZIONE

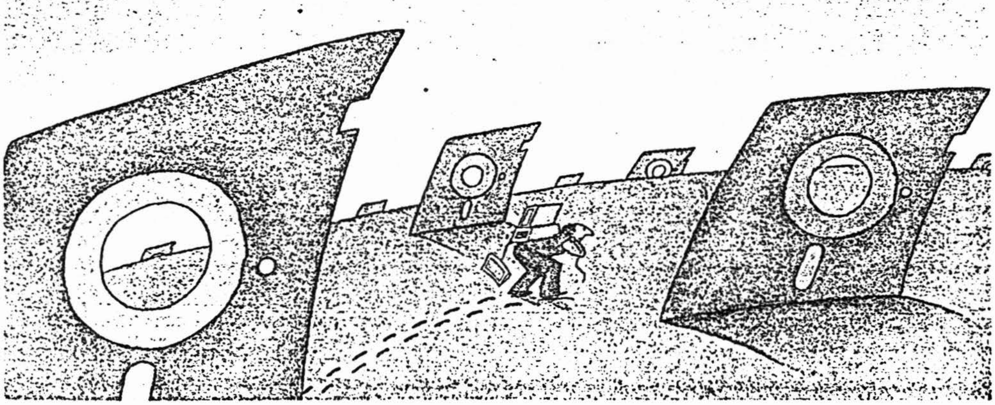
Quando si acquista un modem la prima cosa che si pensa di fare è quella di scambiare programmi e messaggi con amici telematici e di collegarsi a banche dati e BBS. I problemi iniziano all'arrivo delle prime, esagerate bollette della Sip. Nei casi migliori, tutto si risolve con una litigata con i propri genitori o con la consorte; quelli più drammatici finiscono irrimediabilmente con la rottura premeditata del prezioso apparecchio telematico.

Per poter continuare a trasferire kilobyte via modem esiste allora un'unica soluzione: quella di rivolgersi ai programmi compattatori di files.

Ovviamente più un file è corto, più veloce sarà il suo trasferimento, meno scatti verranno conteggiati dalla Sip. Utilizzando un compattatore, è anche possibile ottimizzare e sfruttare al massimo lo spazio sui dischetti e sull'hard disk compattando quei files che usiamo raramente, per decompattarli ogni volta che ci servono.

Il funzionamento di un compattatore è molto complesso e richiede una buona conoscenza della programmazione in generale per essere compreso. Il procedimento di base, tuttavia, è molto semplice: il programma non fà altro che cercare, all'interno del file da compattare, sequenze di caratteri uguali e le sostituisce con un codice più corto che indica il numero ed il tipo di caratteri che ha sostituito.

A questo punto conviene fare una distinzione fra i compattatori. Esistono infatti compattatori (detti crunchers) che riducono la lunghezza in bytes di un programma, lasciandolo eseguibile, ed altri che invece compattano files o interi dischi che, per poter poi funzionare, devono subire prima un processo di decompattazione. L'utilità dei primi sta nel non dover decompattare i files ogni volta che li si vuole eseguire o esaminare, mentre la forza dei secondi sta nel maggior grado di compattazione. Un inconveniente dei crunchers, è l'irreversibilità del loro processo; cioè un file compattato non può essere riportato alla sua forma originaria.



Da ricordare, infine, che la compattazione di un file può rappresentare un ottimo metodo per proteggere i nostri dati da occhi indiscreti.

I COMPATTATORI

Il software di pubblico dominio e quello shareware offrono quattro programmi di archivio/compattazione, tutti eseguibili da Cli: ARC, PAK, ZOO e WARP, ciascuno con proprie caratteristiche e sintassi.

ARC

Disponibile per la prima volta sul FISH DISK #70, ARC può considerarsi il capostipite di tutti i programmi di compattazione/archivio. E' disponibile nelle versioni per tutti i microcomputer, IBM, ATARI, APPLE, ed AMIGA. Ciò permette di trasferire facilmente files in formato ARC da un sistema all'altro.

Il programma è semplice da utilizzare e permette di creare e gestire un archivio di files, ossia un insieme di files riuniti in un unico blocco.
La sintassi di ARC):

ARC (funzione) (nomearchivio.arc) (nomefile)

Il parametro *funzione* è uno dei tanti comandi di ARC; solo quattro di essi vengono comunemente usati:

A aggiunge un file all'archivio;
D cancella un file dall'archivio;
L visualizza la lista dei files nell'archivio;
X estrae un file dall'archivio (decompattandolo).

Per conoscere la lista completa dei comandi di ARC, basta digitare la stringa ARC senza alcun parametro, da Cli.

Il secondo parametro è il nome dell'archivio, seguito sempre dal suffisso .arc, che viene aggiunto automaticamente dal programma quando viene creato un nuovo archivio. Per mantenere la compatibilità con gli altri sistemi operativi, ARC limita i nomi dei files a 16 caratteri. Se vogliamo caricare un file chiamato AMIGA.ARC, abbiamo a disposizione diverse opzioni per estrarre e salvare il file che esso contiene:

ARC X amiga.arc

Se vogliamo salvare il file su un altro disco:

ARC X amiga.arc DF1: nuovo nome

Per estrarre da un archivio chiamato MAX, un file di nome NIC, si deve digitare:

ARC X max.arc NIC

Creare un archivio non è molto più difficile che estrarre un file da uno già esistente. Supponiamo di voler creare un archivio di nome MAX in df1:

ARC A DF1: max.arc

Aggiungere file a questo archivio è ancora piu semplice:

ARC A DF1: max.arc nomefile

Per cancellare un file dall'archivio MAX:

ARC D DF1: max.arc nomefile

Per vedere il contenuto di un archivio, bisogna digitare:

ARC L (DF1:) max.arc

PAK

E' il piu semplice programma di compattazione. La caratteristica più interessante di PAK è quella di creare file auto-decompattanti. Ossia per riportare un file alla sua forma originaria basta mandarlo in esecuzione. C'è pero un piccolo inconveniente: il programma include nel codice compattato la routine per l'estrazione automatica; questo codice extra incrementa così la lunghezza del file compattato. La sintassi di PAK è:

PAK (archivio.pak) nomefile

Il parametro *(archivio.pak)* è il nome del file compattato. Se non c'è l'estensione .pak, quel nome viene inteso come un file da aggiungere all'archivio. E' possibile specificare un percorso per ogni nome di file. Per esempio, per creare un archivio in RAM contenente un file chiamato NIC, bisogna digitare:

PAK RAM: archivio.pak NIC

Per estrarre tutti i files da un archivio PAK, si deve usare il comando CD per entrare nella directory corrente, e mandare in esecuzione il file. Se ad esempio vogliamo estrarre tutti i files da un archivio chiamato max.pak presente in Ram, e salvarli su un disco presente in DF1, bisogna digitare:

CD DF1: RAM max.pak

ZOO

Zoo, disponibile sul Fish disk #164, è molto simile ad ARC, ma offre più flessibilità nel manipolare i files all'interno di un archivio. Diversamente da Arc, Zoo permette di usare nomi di files più lunghi di 16 caratteri e può anche creare una subdirectory in cui inserire i files estratti dall'archivio, usando l'opzione slash (/).
La sintassi di Zoo è:

ZOO (funzione) (archivio.zoo) (nomefile)

Le funzioni essenziali sono:

a per aggiungere un file all'archivio;
d per cancellare un file dall'archivio;
l per avere la lista di tutti i files contenuti in un archivio;
x per estrarre files;

h per visualizzare la lista completa dei comandi.

Bisogna stare molto attenti ad utilizzare questi comandi, poiché Zoo è molto sensibile agli errori. Tuttavia manipolare files con esso è facilissimo. Ad esempio, per estrarre tutti i files da un archivio max.zoo presente su un dischetto posto in DF0, in una directory chiamata NIC, bisogna digitare:

CD NIC ZOO X DF0: max.zoo

Per estrarre un singolo file chiamato JEFF dall'archivio max.zoo, bisogna digitare:

ZOO X max.zoo jeff

Un modo più veloce per decompattare i files in formato Zoo è quello di utilizzare il programma BOOZ (disponibile in molti network e bbs), la cui sintassi è:

BOOZ (funzione) (archivio.zoo) (nomefile)

La funzione **X** estrae uno specifico file da un archivio.

WARP

Mentre, ARC, PAK e ZOO possono archiviare solo files singoli, WARP compatta il contenuto di interi dischi, blocco per blocco, in un singolo file. Uno dei vantaggi di questo metodo di archiviazione è che quando questo file viene decompresso, il disco risultante è esattamente uguale all'originale. Basta solo installarlo da Cli ed il gioco è fatto. Warp, inoltre, identifica il virus SCA sul disco sorgente ed evita di inserirlo in archivio; peccato che non faccia lo stesso anche con gli altri virus.

I possessori delle schede Bridgeboard XT o AT apprezzeranno la capacità di Warp di compattare dischi in formato MS-DOS.

La sintassi per creare un file Warp è:

WARP read start_track end_track archivio.wrp nocomp nozero append mfm +textfile

Per scompattare un file Warp la sintassi è:

WARP write filename1 filename2 filename3

La *start_track* è il numero della traccia del disco da cui deve iniziare la compattazione (per un intero disco si deve porre a 0), e la *end_track* è la traccia finale (per un disco intero deve essere 79). *archivio.wrp* è il nome dell'archivio. E' possibile includere un pathname. L'opzione *nocomp* fà sì che il disco venga archiviato, ma non compresso. Le specifiche *nozero* e *mfm* servono per i dischi non in formato AmigaDOS. La specifica *append* serve per aggiungere dati ad un archivio Warp gia esistente. Con *+textfile* possiamo designare il path ed il nome di un textfile che deve essere visualizzato quando il file viene estratto. I vari parametri sono semplici da capire, ad esempio per compattare un intero disco presente nel drive DF0 in un singolo file in DF1, bisogna digitare:

WARP read 0 79 DF1: nomedisco.wrp

Per archiviare un intero disco in MS-DOS in DF1, in un file WARP in DF0, si deve digitare:

WARP read 0 79DF1: filename.wrp mfm

Per compattare un intero disco in formato Amiga posto in DF0 in un singolo file in DF1, bisogna digitare:

WARP read 0 79 DF1: myfile.wrp

Se vogliamo compattare i nostri files ed i nostri dischi avvalendoci dell'interfaccia INTUITION, tralasciando così la digitazione dei comandi da Cli, abbiamo a disposizione vari tools: ARCHA (un programma di pubblico dominio), DISK MAGIC ed il DISKMASTER. Tutti questi programmi offrono un'interfaccia WIMP per l'archiviazione. **ARCHA** è un semplice programma, che tramite menu a discesa, ci permette di compatttare files e dischi in formato ARC, PAK e ZOO. **DISK MAGIC** (Dmh tecnologies 14873 95th Place Minneapolis, MN 55369 USA $50) e **DISKMASTER** (Progressive Peripherals & Software464 Kalamath st. Denver, CO 80204 USA $50) fanno altrettanto.

Nel prossimo numero parleremo dei CRUNCHERS.



Video Generic Master

L'Amiga è giustamente considerato il personal computer più dotato nel campo del desktop video. Ovvio quindi che, fin dalla sua nascita, le case di software si siano impegnate particolarmente in questo settore.
E' strano, quindi, che solo da poco sia nato un programma che consenta di creare lo scroll di un testo da sovrapporre ad un'immagine, tipo i titoli di coda di un qualunque programma televisivo. Video Generic Master (d'ora in poi VGM) centra questo semplice, ma trascurato, obiettivo senza tanti complimenti: nessun effetto di transizione, nessuno script da editare, solo 4 colori a disposizione per il testo e lo sfondo.

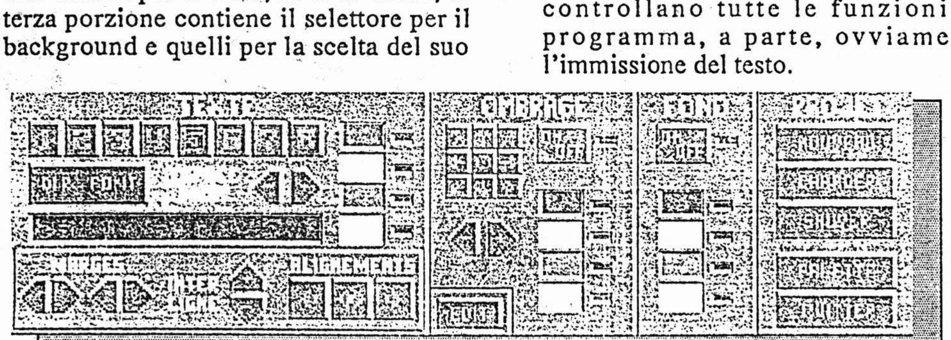
VGM è contenuto su di un disco Workbench: alla partenza apre uno schermo in formato PAL (fondamentale per l'utilizzo qui da noi) su cui si può editare direttamente il testo. Il fondo dello schermo è dedicato al controllo delle funzioni del programma che, infatti, non ha menu a discesa. Questa porzione di schermo è divisa in quattro porzioni: la prima comprende i controlli per il testo (fonts, spaziatura, margini, interlinea e tipo di allineamento); la seconda permette di selezionare il tipo ed il colore dell'ombra per il testo, se la si vuole; la terza porzione contiene il selettore per il background e quelli per la scelta del suo colore. La quarta, infine, è dedicata alle operazioni da disco (caricamento, salvataggio, ecc.) e alla scelta dei 4 colori dalla palette.

E' possibile caricare in RAM fino ad otto fonts diverse da utilizzare per il testo, passando dall'una all'altra con la semplice pressione di un gadget contrassegnato dai numeri da 1 a 8.

Il testo va editato come in un qualsiasi word processor e può raggiungere le 300 linee. Si passa poi a settare i margini, se non lo si è già fatto, a decidere il tipo di allineamento (a destra, a sinistra o centrato), il colore e l'interlinea. Molto sofisticato è il controllo delle ombre: è possibile decidere la posizione di queste in otto direzioni diverse (corrispondenti ad altrettante posizioni della sorgente di luce) ed in più è permesso anche il completo outline, utilissimo per evidenziare il testo quando l'immagine su cui esso si sovrappone tenderebbe a renderlo poco leggibile.

La disponibilità di soli quattro colori, se a prima vista può sembrare una limitazione, si rivela alla fine un vantaggio: troppo spesso si dimentica che l'attenzione dello spettatore è attratta da ciò che legge più che dal colore del testo.

In pratica, con il mouse si controllano tutte le funzioni del programma, a parte, ovviamente, l'immissione del testo.



Video Generic Master

Si accede alla sezione player premendo il tasto EDIT. Avremmo preferito che questa sezione non fosse integrata al VGM, permettendo così di creare delle sequenze di titoli che potessero funzionare senza dover prima caricare il programma principale.

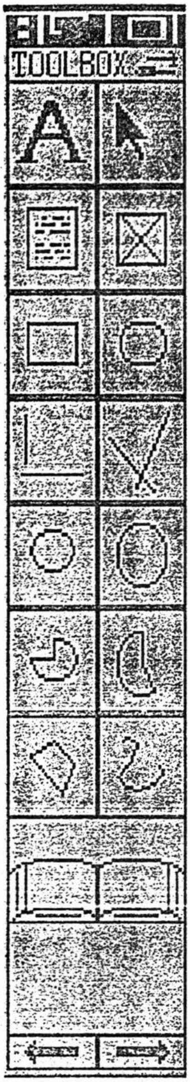
Anche qui le opzioni sono ridotte all'indispensabile: solo quattro tipi di transizione: lo scroll (il testo scorre dal fondo dello schermo verso l'alto), il roll (da destra verso sinistra), una pagina alla volta e una pagina dopo l'altra in scroll. I controlli su queste transizioni sono la velocità e, nel roll, l'altezza sullo schermo della linea di testo.

In quest'ultimo caso è possibile far scorrere il testo, immesso su varie linee nello schermo di lavoro, una linea dopo l'altra, in circolo. Un fastidioso effetto è, però, lo spazio che VGM lascia tra una linea e l'altra e che non si riesce ad eliminare nemmeno immettendo il testo senza lasciare spazi tra una riga e l'altra.

VGM permette di inserire semplici logo su di una linea di testo anche se il procedimento è scoraggiante: bisogna creare un'immagine IFF a due colori, caricare il font editor Calligrapher e sostituire un carattere del font che si userà per il testo con l'immagine creata.

Onestamente meglio lasciar perdere.

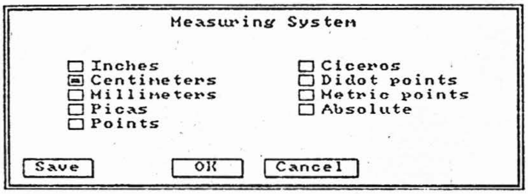
Page Stream



Il sogno di tutti gli utenti Amiga appassionati di desktop publishing è sempre stato quello di vedere sul proprio computer un programma di DTP con caratteristiche simili a quelle dei pacchetti per MS/DOS e Macintosh.
Leggendo le caratteristiche di Page Stream, annunciate almeno sei mesi prima della reale disponibilità del programma, abbiamo tutti pensato che il sogno si sarebbe finalmente avverato.
Page Stream compendia infatti le migliori caratteristiche dei mostri sacri del DTP, Ventura e Pagemaker, adattandole in maniera egregia al mondo Amiga.

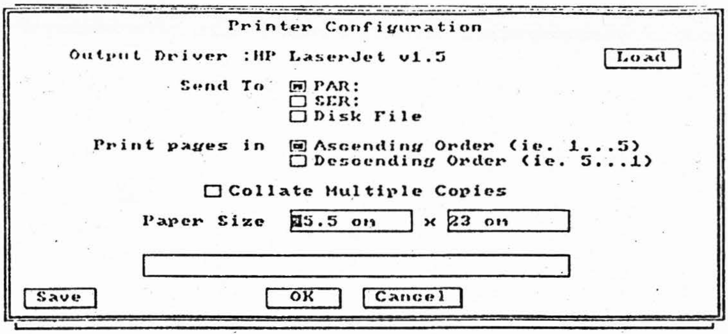
Colpisce subito la grande quantità di opzioni destinate all'uso della grafica object-oriented: oltre ai consueti tools per tracciare rettangoli, riquadri e cerchi, sono presenti strumenti per disegnare archi, archi ellittici e poligoni e per il disegno a mano libera.

Prima di iniziare a lavorare su di un documento, è necessario stabilire una serie di parametri che è possibile memorizzare su disco, associandoli al documento. Da questo punto di vista, Page Stream è perfetto: in una serie di requester che appaiono sullo schermo scegliendo l'opportuna opzione da menu (che vedete riprodotti a corredo di questo articolo) è possibile settare il sistema di misura, il formato della pagina, il tipo di stampante utilizzata. Le possibilità di scelta offerte sono quanto di più completo abbiamo mai visto in un programma di Desktop publishing, non solo per Amiga.



Una volta scelti i parametri, conviene settare le preferenze per la visualizzazione a schermo del documento. Anche qui le possibilità offerte da Page Stream sono numerosissime: dal menu VIEW si stabilisce il rapporto di ingrandimento della pagina, e una serie di opzioni permette di scegliere se vedere o meno sullo schermo i righelli, se evidenziare i bordi dei riquadri situati sulla pagina, se visualizzare o meno le immagini eventualmente presenti, se indicare o meno il concatenamento dei box di testo ed altre ancora.

In tutti i programmi di DTP, per immettere il testo sulla pagina

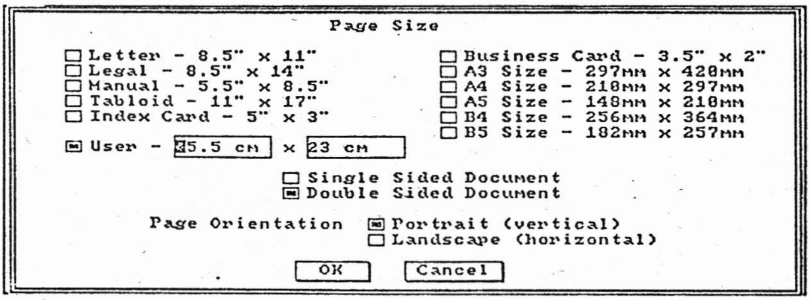


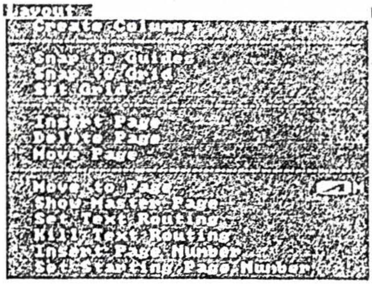
bisogna prima creare un box che lo contenga: Page Stream, volendo, elimina questa necessità: cliccate sull'icona con la lettera A, che, come al solito, seleziona il modo testo, cliccate sul punto della pagina in cui volete che inizi il testo e cominciate a scrivere; finita l'immissione, selezionate col mouse l'icona freccia e poi un punto qualsiasi del testo appena scritto: automaticamente Page Stream circonderà il testo con un box: eccezionale! Oltretutto, in questa maniera, le dimensioni dei fonts sono strettamente legate alle dimensioni del box. Mi spiego meglio: nel modo usuale, cioé se creiamo un box ed immettiamo in esso del testo scegliendo, ad esempio, il font Tymes 11, se poi ingrandiamo il box, il testo occuperà sempre lo stesso spazio all'interno, cioé il font rimarrà sempre Tymes 11. Se, invece, immettiamo direttamente il testo sulla pagina e lasciamo che sia Page Stream a creare automaticamente il box attorno ad esso, quando andiamo ad ingrandire il box anche il font verrà scalato proporzionalmente.

Page Stream, in vero stile Pagemaker, consente di posizionare in qualunque punto della pagina delle guide orizzontali e verticali; questa possibilità, affiancata dalla presenza di una griglia, le cui dimensioni sono modificabili dall'utente, rende il posizionamento dei vari elementi, grafica e testo, di un'estrema facilità.

Da Pagemaker Page Stream prende in prestito anche un altro punto di forza, cioé la tecnica di selezione degli oggetti, addirittura migliorandola quando offre la possibilità di dimensionare non solo il singolo oggetto, ma anche un gruppo di più elementi. Ancora Page Stream offre la possibilità di duplicare i singoli elementi di una pagina determinando le coordinate x e y delle copie e non conosco, al momento, nessun programma di DTP per qualsiasi computer in grado di fare ciò.

Ma, secondo me, è un'altra la caratteristica che rende così piacevole lavorare con Page Stream ed è la gestione dei formati e degli stili del testo. Ancora da Pagemaker Page Stream prende in prestito la possibilità di definire una pagina mastro su cui posizionare il testo e la grafica che vorremo su ogni foglio del documento. Inoltre, Page Stream offre la possibilità di definire dei tags, cio delle "etichette", corrispondenti più o meno ai fogli stile di Ventura e Pagemaker. In pratica si

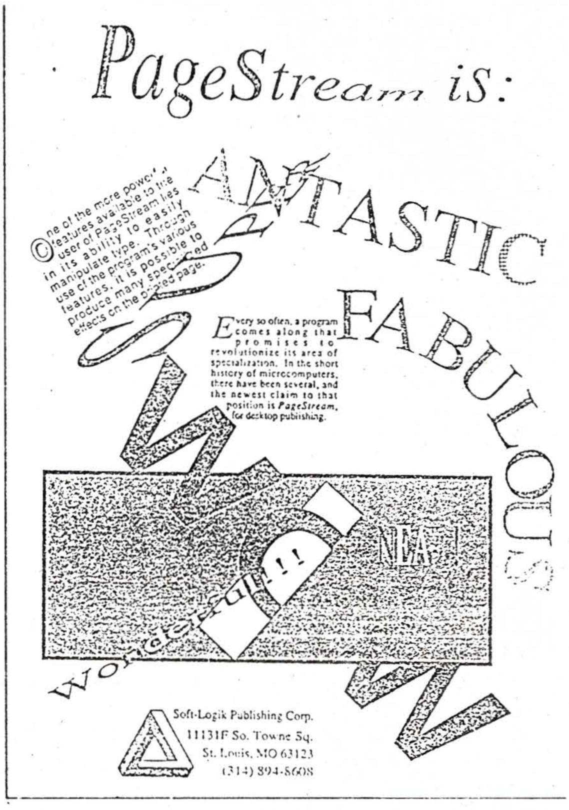




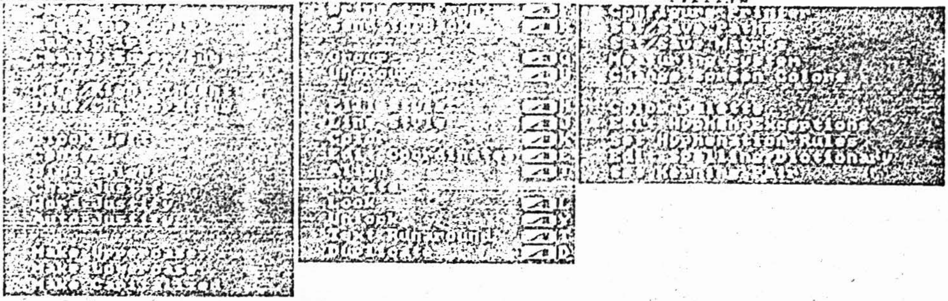




tratta di definire tutti gli attributi del testo (tipo, stile e dimensione del font, spaziatura tra i caratteri e tra le linee, indentazioni, formato del paragrafo, giustificazione del testo, colore e tipo di fill dei caratteri) e salvarli con un nome: in questo modo è possibile richiamare questi parametri in più punti del documento senza dover effettuare ogni volta tutta la serie di settaggi. Un cenno particolare meritano i requester di Page Stream in genere, ma soprattutto quelli che compaiono selezionando una voce dal requester principale "Tags"; è meglio anche qui spiegarci con un esempio: quando vogliamo definire l'interlinea appare un requester in cui è possibile inserire direttamente il numero di punti che si desidera lasciare tra un rigo e l'altro del testo. Mentre si compie questa operazione due righe di testo presenti nel requester si spostano di conseguenza, mostrando l'effetto della spaziatura scelta. Ma non è tutto: provate ad "agganciare" una di queste righe con il mouse e, tenendo premuto il pulsante sinistro, spostatela in su: il relativo numero di punti tra questa riga e quella inferiore verrà calcolato automaticamente e visualizzato: roba da far crepare d'invidia qualunque utente Macintosh.

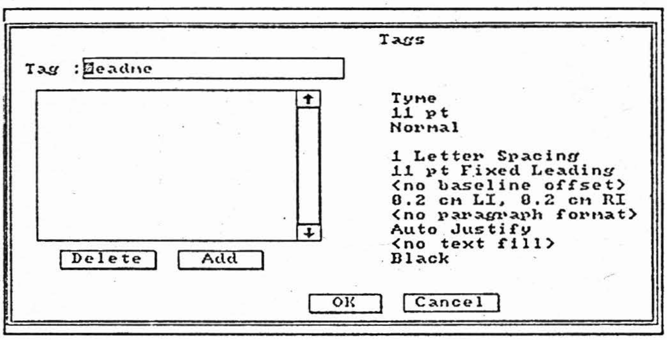


E veniamo adesso ai punti dolenti. Il problema fondamentale che mi si è presentato lavorando a lungo con Page Stream è l'"instabilità" : ogni tanto il programma si "inchioda" e non sempre nelle stesse condizioni, nel senso che non è possibile stabilire un nesso preciso tra determinate operazioni e guru meditations. Ho comunque notato una certa propensione del programma a bloccarsi quando si tenta di importare



immagini a colori. In questo caso l'unica soluzione è quella di resettare l'Amiga, perdendo tutto il lavoro fatto dall'ultimo salvataggio. Quello che sento di consigliarvi è di salvare spesso il vostro lavoro, pratica sempre saggia, ma in questo caso fondamentale.

Page Stream può importare immagini in formati diversi: Aegis Draw Plus, IFF, IFF-ILBM, ATARI Degas, EPS (Encapsulated Postscript) Macintosh, IMG IBM, mentre per il testo si limita al formato WordPerfect e ASCII. Il manuale non si dilunga molto sulle tecniche di importazione dei files per cui a molti problemi bisogna cercare di trovare una qualche soluzione procedendo per tentativi. Spesso Page Stream si blocca



irrimediabilmente quando si tenta di importare un particolare brush da DeluxePaint III, mentre con altri brush tutto funziona perfettamente.

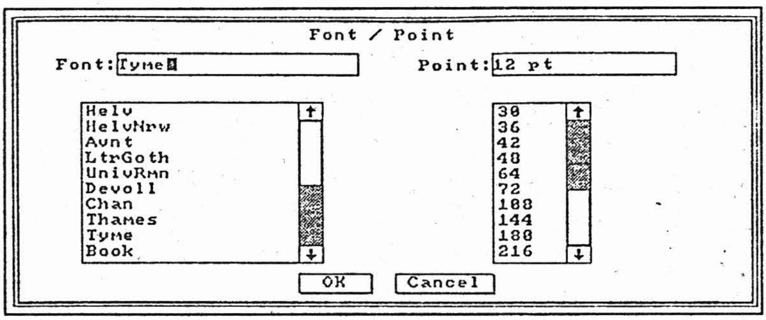
Grosso problema si è poi rivelata l'importazione del testo in cui erano presenti lettere accentate: ho provato con numerosi word processors, tra cui Transcript, C1-Text e perfino CygnusEd Professional, ma immancabilmente, caricato il testo in Page Stream, le vocali accentate erano sempre scomparse. Il caso limite, poi, è rappresentato da Excellence!, word processor che si permette di deridere WordPerfect, e non è, poi, in grado di riconoscere nemmeno il testo salvato da se stesso. Provate a scrivere un testo con la tastiera italiana, inserendo quindi vocali accentate; salvate il testo e, subito dopo, ricaricatelo da disco: invece delle vocali accentate troverete dei simboli incomprensibili. Indegno per un programma che, in Italia, viene venduto al "modico" prezzo di 450.000 lire!

Ma torniamo a Page Stream. Ancora da migliorare è senz'altro il refresh dello schermo: molto spesso alcune modifiche non vengono

visualizzate fino a quando non si cambia ingrandimento e quindi lo schermo viene completamente aggiornato. Capita così che in alcuni momenti abbiamo sullo schermo due-tre cursori per il testo, di cui solo uno è quello reale. Lo stesso capita sui righelli che, in alcuni momenti, sono invasi di linee non corrispondenti alla posizione del pointer.

Infine, una scelta strana quella della Soft-Logik di un proprio formato per i fonts: se da un lato ciò consente di ottenere un'eccezionale output su stampanti ad aghi, limita notevolmente le possibilità dei possessori di laser con Postscript di utilizzare le fonts di questo linguaggio, nettamente superiori come qualità a quelle fornite dal programma.

In definitiva, se avete fretta di provare questo mostro del DTP, compratelo subito, imparate ad usarlo ed aspettate l'upgrade che verrà offerto a condizioni privilegiate ai possessori di questa versione (che, per inciso, la 1.6). Se invece non avete fretta, aspettate che la Soft-Logik abbia finito di "ripulire" Page Stream dai numerosi bugs e di ampliarne



ancora di più le caratteristiche (ci hanno detto che sono già al lavoro).

A quel punto, statene certi, avrete il miglior programma di DTP esistente.

Rimane l'amarezza di aver dovuto acquistare il programma in Inghilterra e di averlo pagato, al cambio, 570.000 lire. Se tale cifra sembrerà una bazzecola agli utenti IBM e Macintosh, che per Ventura e Pagemaker devono sborsare un cifra almeno tre volte maggiore, d'altra parte ci dà fastidio il fatto che venga messo in commercio un programma non perfettamente funzionante e che all'utente che lo ha acquistato venga risposto di aspettare la nuova versione.

PAGE STREAM 1.6
Soft-Logik
11131F S. Towne Square
St. Louis, MO 63123

Ventura e Pagemaker sono marchi registrati rispettivamente della Rank Xerox e della Aldus

B.A.D.

Come velocizzare i nostri dischi

L'AmigaDOS salva i files nelle zone libere di un disco: è facile intuire che, scrivendo e cancellando varie volte sullo stesso disco, le zone libere non saranno sempre contigue. Ciò comporta che un file può essere suddiviso in blocchi non sequenziali. In ogni caso il sistema operativo di Amiga è in grado di leggere correttamente il file in quanto ogni blocco di dati sa dove si trova il blocco precedente ed il successivo: lo svantaggio è che il caricamento in memoria di un file molto frammentato richiede più tempo.

Numerosi programmi, anche di pubblico dominio, possono risolvere questo problema, riorganizzando la disposizione dei dati sui dischi. Uno dei migliori è B.A.D. (Blitz Amiga Disk). Gli autori del programma affermano che il caricamento di un disco "trattato" con BAD, subisce un incremento di velocità del 500%. BAD, in effetti, non fà altro che copiare il contenuto del disco sorgente in memoria e poi riorganizzarlo sul disco destinazione. BAD può essere utilizzato anche per gli hard disk in cui le conseguenze della frammentazione dei dati, parlando di 20 o più mega di capacità in confronto agli 880k di un dischetto, sono molto più evidenti.

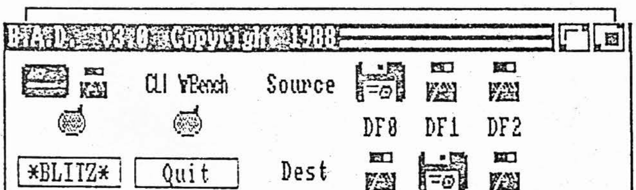
Come funziona il programma

BAD parte da workbench. Cliccando sull'icona del programma appare, dopo i crediti di rito, una finestra con vari gadgets. Il gadget BLITZ serve per selezionare il dispositivo sorgente, hard disk o floppy disk. Affianco troviamo il gadget che serve per decidere se vogliamo ottimizzare un disco per l'uso da worbench o da Cli.

WORKBENCH MODE: l'intero disco, incluse le directories e le subdirectories, viene ottimizzato con una speciale preferenza per i files che hanno l'estensione ".info", ossia le icone: in questo modo il contenuto delle finestre appare più velocemente sullo schermo del workbench.

CLI MODE: in questo caso i files vengono riorganizzati in modo da rendere il più veloce possibile il funzionamento di comandi quali DIR, LIST.

Fatte tutte le scelte circa i drives ed i modi operativi, inseriamo i



dischi nel drives e clicchiamo sul gadget BLITZ. BAD comincerà a lavorare: il tempo necessario per l'ottimizzazione del disco varia a seconda della quantità dei dati presenti sul disco. Quando BAD avrà finito inseriamo la copia ottenuta con la protezione abbassata, in modo da permettere al worbench di riscrivere l'.info file aggiornato.

E' possibile utilizzare BAD anche con un solo drive: ovviamente tutta l'operazione risulterà notevolmente più lenta.

Per ottimizzare e velocizzare un disco, avendo due drives, è richiesto un minimo di 370k di Ram libera. Per questa ragione se si hanno soli 512k bisogna lanciare il programma da Cli. Il disco da trattare, inoltre, deve essere privo di errori, non protetto dalla copia.

BAD può essere richiesto direttamente a:

MARK HELLMANN
P.O. BOX 1112 WHEATRIDGE, CO 80034-1112 -USA-
BAD è stato scritto da Mark Hellmann e Reid Bishop.

i files .info

I files ".info" sono creati dal workbench per ogni directory a cui deve accedere. Questo, ad esempio, è il modo in cui viene velocizzato l'accesso ai files .info usati per le icone. L'aggiunta di un file ad una directory costringerà il Workbench a rileggere l'intera directory e poi a riscrivere l'info.file per essa. Questo tipo di accesso rallenta notevolmente le operazioni.

Il caricamento dei dati e dei programmi viene rallentato anche per altre ragioni, come l'accesso ad un disco con una differente versione del Worbench; quest'ultima, infatti, deve riscrivere l'intero info.file con il suo numero di identificazione. Un'altra caso è quando il clock del sistema è settato con una data meno aggiornata di quella presente sul disco. Una possibile soluzione consiste nel riorganizzare i files in directory e subdirectory, per raggiungere la massima velocità di caricamento.

Se notiamo che la visualizzazione del contenuto di un drawer richiede molto tempo, dobbiamo controllare che il nostro disco abbia la protezione abbassata . Ciò permetterà al Workbench di riscrivere il file .info, aggiornandolo.

Pregi e difetti di un non-linguaggio

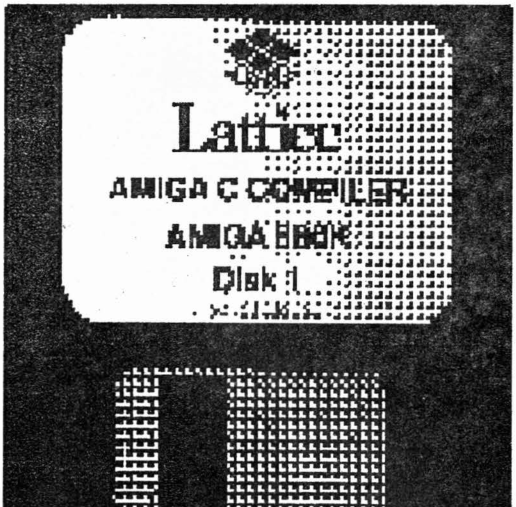
Comunemente si definisce linguaggio un insieme di codici che fungono da tramite fra l'utente ed il computer, in modo che siano sufficientemente simili al linguaggio umano e contemporaneamente comprensibili al calcolatore tramite un programma interprete o compilatore (ricordate quello che dicemmo nello scorso numero?).

I linguaggi piú comuni sono molto piú simili al linguaggio umano che non a quello del calcolatore; infatti, citando il BASIC come esempio, sono utilizzati termini comuni al linguaggio umano (inglese); PRINT e WRITE significano "stampa" e "scrivi", AND, OR e NOT significano "e (congiunzione)","o" e "non". L'utilizzo di termini "umani" rende molto facili e comprensibili questi linguaggi (COBOL in primis) ma li penalizza in velocitá e potenza.

Sono nati cosí alcuni linguaggi facilmente comprensibili al calcolatore (il che si traduce in velocitá e potenza), ma non troppo facilmente comprensibili all'uomo.

Il primo di questi linguaggi fu l'Algol 60 (1960), presto soppiantato dal CPL (1963). Questi due linguaggi non erano peró sufficientemente vicini al linguaggio macchina per consentirne un vantaggioso utilizzo.

Nel 1967 Martin Richards progettó a Cambridge il Basic CPL (poi BCPL) che era peró un linguaggio troppo legato alla macchina su cui era stato progettato e quindi non facilmente trasportabile ed utilizzabile su altri sistemi.



Nel 1970 Ken Thompson (che lavorava ai Bell Laboratories su sistemi UNIX) progettó il linguaggio "B"; creato da un professionista, il B era peró un linguaggio molto difficile da utilizzare e troppo poco simile al linguaggio umano per poter essere utilizzato per grossi programmi.

Due anni dopo, nel 1972, sempre ai Bell Labs, Dennis Ritchie progettó un nuovo linguaggio, potente come il B, ma molto piú facile da utilizzare: dalla lettera seguente la B, questo

linguaggio prese il nome di "C". Anche il "C" é ancora troppo poco simile alla lingua umana per poter essere definito linguaggio. Ha peró un vantaggio rispetto ai suoi predecessori: la possibilitá di essere utilizzato in molte applicazioni, particolaritá che non avevano né l'Algol o il CPL (troppo dedicati ad un utilizzo scientifico) né il BCPL o il B (dedicati ad applicazioni di sistema).

Il C é standard e puó essere facilmente utilizzato sia nell'una che nell'altra applicazione senza limitazioni; inoltre, nella peggiore delle ipotesi, si presta benissimo ad essere utilizzato insieme al linguaggio macchina: capita infatti molto spesso di vedere programmi scritti parte in C e parte in codice macchina.

Tutte queste caratteristiche giustificano l'enorme successo del nuovo linguaggio al punto che sui sistemi UNIX il 95% dei programmi e parte del sistema operativo sono scritti in C. Inoltre la facilitá di comprensione da parte della macchina ha permesso la rapida diffusione del C anche su altri sistemi (CPM, MS-DOS) in cui é oggi utilizzatissimo, anche se in misura minore rispetto ai sistemi UNIX (ricordiamo che l'Amiga non é uno UNIX ma é molto simile a questo).

Cosa occorre per programmare in C

Per programmare in C sono necessari:

un Editor (per iniziare va benissimo il comando ED del workbench)

il compilatore C (gli esempi riportati in questi articoli sono relativi al Lattice C Compiler v4.0 o superiore)

Consigliamo inoltre l'uso di uno Shell (ConMan o quello del WB 1.3) e quello del PopCLI (un'utility che apre un nuovo CLI premendo i tasti Amiga-Sinistro ed ESC); entrambi questi strumenti sono forniti con il Lattice C v4.0 (o superiore) o reperibili in librerie di pubblico dominio (FishDisk #35).

Come avviene la creazione di un programma

1) Viene editato il codice sorgente IL CUI NOME DEVE TERMINARE CON IL SUFFISSO ".C" con un comando del tipo "ED Test.c" (nel Lattice C v5.0 si usa "LSE Test.c").

2) Il file viene salvato con un'apposita opzione dell'editor (ESC e poi X e RETURN per l'editor del WorkBench)

3) Viene avviata la compilazione, che si divide in 4 fasi:

a) Test di sintassi (per verificare se il programma é corretto
b) Prima fase di compilazione (con generazione di un secondo file il cui nome termina con ".Q" (Lattice C) o ".asm" (Aztec C ed altri)
c) Seconda fase di compilazione (con generazione di un secondo file il cui nome termina con ".O")

Le fasi 1 e 2 vengono ripetute per ogni modulo del programma (in C é infatti possibile suddividere un programma in moduli piú piccoli che verranno poi fusi assieme)

4) Fusione (Linking) dei vari moduli e delle librerie (la porzione del compilatore che verrá inclusa nel programma per permetterne il funzionamento)

5) Prova del programma appena ottenuto

Un errore in una qualsiasi di queste fasi comporta la ripetizione del procedimento (ovviamente verrá rieditato e ricompilato solo il modulo difettoso, mentre dovranno essere ripetute le fasi 4 e 5).

Traducendo in termini di CLI le varie fasi di cui sopra si ottengono con:

1 e 2) **ED Prog.c**

3a-b) **LC1 Prog.c**

3c) **LC2 Prog.q**

4) **BLINK FROM LIB:C.o+Prog.o TO Prog LIBRARY LIB:amiga.lib+LIB:lc.lib [+LIB:lcm.lib]**

Nota: L'ultima aggiunta é necessaria solo se il vostro programma utilizza numeri non-interi. Comunque nulla accade (se non un rallentamento del linking) se la si aggiunge anche nel linking di un programma che non utilizza i numeri in virgola mobile (non-interi)

5) **PROG**

Questa é la procedura da utilizzare nel caso in cui il vostro

programma sia composto da un solo modulo (il caso piú frequente).

Se invece il programma é composto da piú moduli (che chiameremo ModX, dove X é il numero del modulo) la procedura é la seguente:

1 e 2) ED Mod1.c – ED Mod2.c .... ED ModN.c

3a-b) LC1 Mod1.c – LC1 Mod2.c ..... LC1 ModN.c

3c) LC1 Mod1.q – LC1 Mod2.q ..... LC1 ModN.q

4) BLINK FROM LIB:C.o+Mod1.o+Mod2.o+.....+ModN.o TO Prog LIBRARY LIB:amiga.lib+LIB:lc.lib [+LIB:lcm.lib]

Nota Bene: Le due linee del Passo 4 SONO UN'UNICA LINEA CLI, mentre le altre sono varie linee, separate da -- o .....

5) PROG

Difficile? Non mi sembra poi tanto. Anche perché la Lattice ha fornito, insieme al compilatore C, il tool LC che provvede ad eseguire i passi 3a-b-c e 4 per TUTTI i moduli di un programma. L'uso é semplicissimo:

LC -L Mod1.c Mod2.c Mod3.c ... ModN.c

(Se il vostro programma utilizza solo numeri interi)

LC -Lm Mod1.c Mod2.c Mod3.c ... ModN.c

(Se il vostro programma non utilizza solo numeri interi)

Da notare che le opzioni -L e -Lm indicano a LC di invocare il linker specificando o meno la libreria lcm.lib (che contiene le funzioni per l'utilizzo dei numeri in virgola mobile); non specificando nessuna di queste due opzioni dovrete invocare il linker come detto sopra (passo 4).

Due esempi

E' doveroso quindi proporre due programmi esempio (il primo di un solo modulo ed il secondo di due) che proverete a compilare.

Da questo momento scriveremo in caratteri normali quello che sará scritto sul video dal computer, ed in neretto quello che dovrá scrivere l'utente.

Abbiamo presupposto che il prompt del CLI sia "1>", ma potrebbe anche essere "2>" o altro (non é questo un particolare di grossa importanza).

Esempio numero 1

Caricate il compilatore C, attendete la comparsa del prompt "1>" e, nel caso si tratti del Lattice C v4.0, digitate

ENDCLI

e premete Amiga-sinistro ed ESC per ottenere un nuovo processo CLI dotato di Shell, poi premete F2 per allargare al massimo le dimensioni della finestra. Quindi digitate:

1> **ED MyProgram.c**

(se non avete il Lattice C v5.0) oppure

1> **LSE MyProgram.c**

(se avete il Lattice C v5.0)

Se il dischetto del C non fosse provvisto di un editor, provvedete a copiare il comando ED del WorkBench nella directory c del disco del C con un comando del tipo:

1> **COPY df1:c/ED df0:c**

(supposto che abbiate il WorkBench in df1: ed il primo disco del C in df0:)

Entrati nell'editor digitate il seguente programmino:

```
void main()
{
printf ("Esempio di programma C\n");
}
```

Notate lo shortcut \n (E NON /n) per andare a capo e il

punto e virgola alla fine della terza linea.Fatto questo salvate (con l' apposito menu dell'editor - per il comando ED del WorkBench utilizzare ESC seguito da X e RETURN) e quindi uscite dall'editor.

Riottenuto il prompt, digitate:

1> **LC -L MyProgram**

Ed attendete la compilazione. Se si verifica un errore durante la compilazione rieditate il file e cercate l'errore.
Un eventuale errore durante la fase di linking é segnalato dalla scritta *Undefined symbols -- First referenced* indicante il simbolo che il linker non é riuscito a trovare seguito dal nome del modulo in cui questo simbolo é stato trovato per la prima volta. NON TENTATE DI ESEGUIRE UN PROGRAMMA NELLA CUI FASE DI LINKING SONO STATI GENERATI ERRORI, altrimenti Guru Meditation o blocco del sistema...
Se non ci sono errori potrete digitare:

1> **MyProgram**
Esempio di programma C
1>

Da notare la risposta del computer. Se volete provare a vedere come viene segnalato un errore in fase di linking, cambiate la linea 3 del programma da

printf ("Esempio di programma C\n");

a

boh ("Esempio di programma C\n");

Il linker vi segnalerá questo errore:

Undefined Symbols First referenced _boh MyProgram.o

Che vuol dire che il linker non é riuscito a trovare il simbolo boh e che questo simbolo é stato richiesto per primo dal modulo oggetto MyProgram.o (il modulo oggetto é quello restituito dalla seconda fase di compilazione; quello restituito dalla prima - che verrá poi cancellato durante la seconda fase di compilazione - é detto modulo quad).
Ma passiamo ad analizzare un po' il nostro programma. Diciamo subito che il nostro programma é composto da una sola

funzione. Ma cosa é una funzione? La funzione é composta da una serie di istruzioni raggruppate secondo la seguente sintassi:

NomeDellaFunzione() Notare le parentesi, che indicano che questa é una funzione
{ Parentesi graffa aperta: inizio del blocco di istruzioni
istruzione1; Istruzione SEGUITA DAL PUNTO E VIRGOLA
istruzione2; ....
ultimaistruzione;
} Parentesi graffa chiusa: fine del blocco di istruzioni e della funzione

Le varie istruzioni possono essere operatori di condizione (saranno descritti in seguito - questi ultimi non vogliono il punto e virgola), operatori di assegnamento (anche questi descritti in seguito) oppure chiamate ad altre funzioni. Queste saranno eseguite in ordine sequenziale, una dopo l'altra, cosí come sono state scritte. La chiamata ad una funzione avviene con

Funzione();

Se la funzione non richiede parametri

Funzione (Parametro);

Se la funzione richiede un solo parametro

Funzione (Parametro1,Parametro2,...,ParametroN);

Se la funzione richiede piú di un parametro

I parametri sono i dati che la funzione chiamante passa come argomenti su cui lavorare alla funzione chiamata. Da notare che anche printf() é una funzione (contenuta nella libreria lc.lib). La funzione che sará chiamata per prima, al lancio del programma DEVE avere come nome main.

Un importante particolare: IN C NON CONTANO NE' SPAZIATURE, NE' A-CAPO, NE' TABULAZIONI, MA MAIUSCOLE E MINUSCOLE SONO DIFFERENTI: cosí main() é diverso da Main() e da MAIN(). Inoltre nei nomi di funzioni o di variabili non sono ammessi spazi (al loro posto si usa l'underscore _), ma solo lettere, numeri ed il simbolo _; ogni nome di variabile o funzione deve iniziare con _ oppure una lettera.

Secondo esempio

Digitate:

1> ED Mod1.c

oppure

1> LSE Mod1.c

se avete il Lattice C v5.0 (da ora in poi questa distinzione non verrá piú fatta: quindi se avete il Lattice C v5.0 digitate LSE dove é scritto ED)

Entrati in Editor digitate:

```
void main()
{
printf ("Il quadrato di 4 é %d\n",Quadrato(4));
}
```

Quindi salvate, uscite dall'editor e digitate:

1> ED Mod2.c

e editate la seconda parte del programma: la funzione Quadrato() (useremo sempre le parentesi quando parleremo di una funzione, per distinguerla da una variabile):

```
Quadrato(x)int x;
{
return(x*x);
}
```

Notare l'uso di un parametro: x. La seconda linea specifica che x é una variabile di tipo intero (non decimale o contenente caratteri); ma non abbiamo ancora definito il concetto di variabile: una variabile é costituita da una o piú celle di memoria destinate a contenere un valore. La variabile é identificata da un nome, tramite il quale verrá citata successivamente nel programma, quando serve. E' necessario specificare al compilatore di quale tipo di variabile si tratta prima del suo utilizzo. Questa operazione é detta DICHIARAZIONE di una variabile. Parleremo meglio di variabili (e della funzione di libreria printf()) nel prossimo articolo. Per ora continuiamo con il nostro

programma a due moduli.

Per effettuare la compilazione digitare:

1> **LC -L Mod1 Mod2**

Se non si sono verificati errori potrete verificare il funzionamento del programma con:

1> **Mod1**
Il quadrato di 4 é 16
1>

Quando esiste piú di un modulo sorgente (quello che contiene i codici sorgenti del programma ed il cui nome termina con .C), il programma finale prenderá il nome del primo modulo specificato. Notate che LC accetta il nome del modulo con o senza l'aggiunta .C.

Provando a compilare solo l'uno o l'altro modulo il linker non riuscirá a trovare le funzioni Quadrato() e main() rispettivamente. Il linker ANTEPONE UN UNDERCORE DINANZI AD OGNI SIMBOLO (NOME DI FUNZIONE O DI VARIABILE), quindi main sará segnalato come _main (non usiamo parentesi perché si tratta del simbolo main e non della funzione main().

Ora proviamo ad unire i due files sorgenti in uno solo tramite il comando JOIN del WorkBench. Inseriamo tale dischetto in DF0: e digitiamo:

1> **DF0:c/JOIN Mod1.c Mod2.c AS Full.c**

Riponiamo il disco del C in DF0: e quindi digitiamo:

1> **LC -L Full**
1> **Full**
Il quadrato di 4 é 16
1>

Il funzionamento del programma é rimasto immutato. Ovviamente non conviene suddividere programmi piccoli come quello visto sopra, ma quando si hanno lunghissimi programmi che richiedono vari minuti per essere compilati, conviene la suddivisione in moduli che, in programmi piú complessi, richiede qualche accorgimento in piú. Tutto questo sará descritto in seguito.

Ritornando al nostro secondo programma notiamo l'uso di return(). Sebbene possa sembrare una funzione sotto tutti gli

aspetti, questa é una pseudo-funzione che esce dalla funzione in cui é citata ritornando il valore contenuto tra parentesi. Usata senza parentesi esce dalla funzione senza ritornare alcun valore, proprio come se si fosse raggiunta la parentesi graffa chiusa che delimita la fine della funzione. Se una funzione non ritorna alcun valore deve essere dichiarata come "void".

La dichiarazione di una funzione avviene in maniera simile a quella di una variabile: la dichiarazione avviene prima del nome della funzione stessa, ma in questo caso non é obbligatoria: il default é int. Quindi scrivere Quadrato(x) é come scrivere int Quadrato(x), anche se questa operazione é svolta di default dal compilatore stesso. Il non dichiarare come void una funzione che non ritorna alcun valore non costituisce errore tanto grave da compromettere la compilazione di un file, quindi il compilatore fará comparire sul video la scritta

Warning: Function returns value mismatch.

Per terminare questa prima puntata di corso di C vorrei descrivere brevemente come il compilatore segnala gli errori; di ogni errore viene segnalato prima il modulo che l'ha generato, poi il numero della linea del programma in cui si é verificato, quindi il tipo (Error = errore, Warning = avvertimento da parte del compilatore) ed un numero indicante il tipo di errore, infine una breve descrizione. Negli ultimi compilatori prima dell'avviso di errore compare stampata sul video la linea che ha generato l'errore, con evidenziato il punto che ha generato l'errore stesso (una freccia ^ nel Lattice C v4.0 e con l'inversione in rosso con il Lattice C v5.0).

Da notare che alcuni errori (come il numero 57: semi-colon expected - che indica che si é saltato il punto e virgola alla fine di una linea) sono fatali su alcune versioni del compilatore C (v4.0 o inferiori) e non fatali nelle versioni successive.

Terminiamo questo primo articolo sul C: nel prossimo articolo verranno meglio descritte le variabili, le funzioni e la funzione di libreria printf() - che, spero l'abbiate capito da soli, serve a inviare dati sul video.

Come descritto nell'articolo sul numero scorso, il "vero" linguaggio del computer é il linguaggio macchina, sebbene ogni computer che si rispetti abbia un linguaggio "dedicato": per citare alcuni esempi, i sistemi operativi MS-DOS erano dedicati al linguaggio PASCAL, mentre gli UNIX (cui l'AmigaDOS é molto simile) sono dedicati al linguaggio C; i vecchi sistemi che non disponevano di DOS (come il buon vecchio ZX Spectrum o il Commodore 64) erano dedicati al BASIC, tramite il quale si poteva accedere al codice macchina. Non mancano peró alcune eccezioni: sto parlando di drammatici casi, come quello del CASIO PB700, che disponevano (oltre che di pochissima memoria) solo del linguaggio BASIC, e non era assolutamente possibile accedere al linguaggio macchina in nessuna maniera: ovviamente su questi computer i risultati ottenibili erano piuttosto scarsi. Infatti il linguaggio macchina é utilizzato in quei casi in cui é necessaria o un'elevata velocitá (si pensi, per esempio ad un caricatore IFF: in linguaggio macchina é piú veloce di un buon 75% di un suo equivalente C) o in cui é necessario un diretto accesso all'hardware della macchina (per esempio, un caricatore per dischi formattati secondo altri standards: si veda il Dos2Dos).

Per ottenere quindi il massimo delle prestazioni da un computer, appare evidente che il linguaggio macchina (o l'Assembler - consultare il precedente articolo di questo corso) é la scelta obbligata.

Un programma scritto in linguaggio macchina, peró, é legato in modo molto stretto alla macchina su cui é stato generato: per cui se é estremamente facile trasportare programmi scritti in altri linguaggi fra macchine simili (UNIX - Amiga, Amiga - Atari ST), la cosa si complica non poco se si deve trasportare un programma in codice macchina da un computer ad un altro se questi hanno lo stesso microprocessore (detto anche CPU) e diventa addirittura drammatica nel caso in cui le due macchine non dispongano di CPU della stessa famiglia.

Le principali caratteristiche di un microprocessore

Analizziamo ora le principali caratteristiche delle varie CPU presenti oggi in commercio.

Innanzitutto la larghezza del BUS DATI, che puó essere a 8, 16, 32 o 64 bits. Praticamente la larghezza del DATA BUS indica la larghezza dei numeri che possono essere gestiti dal

microprocessore; esempio pratico: immaginate due calcolatrici, la prima che disponga di sole 8 cifre e la seconda che disponga di 12 cifre: é ovvio che la seconda calcolatrice potrá gestire numeri molto piú grandi della prima, e sará probabilmente piú sofisticata e capace di svolgere calcoli piú complessi della prima. La stessa cosa accade con i microprocessori: un microprocessore che abbia il bus dati di 8 bit puó gestire numeri che vanno da 0 a (2^8)-1=255, mentre un microprocessore che abbia 16 bit puó gestire numeri che vanno da 0 a (2^16)-1=65535. Normalmente, peró un processore che abbia 16 bit puó lavorare sia a 16 che ad 8 bit, come un processore a 32 bit puó lavorare a 32, 16 o 8 bit: per esempio nella famiglia dei 680xx (cui appartiene il processore dell'Amiga) si puó scegliere la larghezza del dato su cui lavorare specificando, al termine dell'istruzione in questione, uno dei suffissi .B (byte = 8 bit), .W (word = 16 bit) o .L (long word = 32 bit).

Seconda caratteristica di un microprocessore é la FREQUENZA DI CLOCK: infatti ogni microprocessore riceve dall'esterno un segnale (detto appunto segnale di clock) che la CPU utilizza come riferimento per la velocitá delle operazioni da compiere e per indicare l'inizio o la fine di una operazione. Si badi bene peró che un processore che abbia una frequenza di clock di 4 MHz (4000000 di impulsi al secondo) non effettua altrettante istruzioni in un secondo, perchê le varie istruzioni hanno differente durata, che nel caso piú breve non é mai inferiore a 2 cicli di clock: sempre citando come esempio il processore dell'Amiga, un movimento veloce in un registro richiede 4 cicli di clock (4 impulsi) per essere portato a termine, mentre una divisione a 32 bit ne richiede 145.

Terza caratteristica principale di un microprocessore é il numero di interrupts che il microprocessore puó gestire: tecnicamente si definisce interrupt un procedimento tramite il quale un dispositivo esterno puó richiamare l'attenzione del microprocessore su un evento verificatosi: per esempio il disk drive puó segnalare al microprocessore l'inserimento o l'estrazione di un disco dal drive o la fine del trasferimento di dati semplicemente inviando un segnale ad un piedino del microprocessore (il piedino di segnalazione di interrupt-in-arrivo) che provvederá prima a riconoscere quale dispositivo ha generato l'interrupt e quindi a prendere i provvedimenti del caso.

Quarta caratteristica di un microprocessore é la quantitá di memoria indirizzabile senza particolari accorgimenti: in pratica la

quantitá di memoria a disposizione del microprocessore.

Quinta caratteristica di un microprocessore é il numero di registri (le unitá basilari interne nelle quali il processore compie le operazioni): ovviamente piú sono, meglio é.

Sesta caratteristica é la gestione interna degli errori: infatti, un errore in un programma in codice macchina porta spesso ad un crash del sistema, che per giunta non é nemmeno in grado di segnalare all'utente la causa dell'errore. Alcuni microprocessori (tra cui quello dell'Amiga) dispongono di un sofisticato sistema di gestione degli errori: quindi un errore in un programma porta spesso al blocco del solo programma in cui si é verificato l'errore, mentre gli altri programmi in funzione in quel momento continuano a girare senza problemi: é quello che in Amiga accade quando compare il requester "Software Error - Task Held". Solo in casi molto drastici l'errore porta ad un blocco del sistema, che é peró in grado di segnalare come, dove e perchè si é generato l'errore (sto parlando dei due numeri che compaiono dopo una Guru Meditation).

Settima caratteristica é il set di istruzioni, ossia la serie di istruzioni (ed il loro corrispondente codice binario) con le quali il processore lavora. Generalmente i vari processori hanno set di istruzioni nettamente diversi, ma esistono "famiglie" di processori che hanno lo stesso set di istruzioni in comune. Esistono due tipi di famiglie di processori: quelle in cui i processori si differenziano solo per velocitá (si vedano gli Zilog Z80, Z80A, Z80B, che hanno uguale circuiteria, ma differente frequenza di clock - 2, 4. 5, 6 MHz) e quelle in cui i processori hanno differenze anche nella piedinatura e lievissime differenze nel set di istruzioni. E' il caso questo della famiglia di microprocessori cui appartiene il processore dell'Amiga.

Ottava caratteristica é la larghezza del BUS DATI ESTERNO: infatti, sebbene il microprocessore abbia un bus dati largo, per praticitá, quando deve inviare dati ad un dispositivo esterno, puó inviare il dato a piú riprese, suddividendolo in frammenti di larghezza minore: per esempio, un dato di 32 bit puó essere inviato all'esterno in due tempi (prima i 16 bit superiori, poi quelli superiori)

La famiglia dei 680xx

Cervello delle attivitá dell'Amiga é il microprocessore

MC68000, prodotto dalla casa statunitense Motorola. In questo integrato di 64 piedini risiedono circa 68000 transistors piú qualche altra decina di migliaia di componenti passivi, il tutto su una piastra di silicio la cui superficie é inferiore al mezzo centimetro quatrato.

Nell'Amiga, peró, il 68000 viene sfruttato solo fino all'8% delle sue reali possibilitá; altri gravosi compiti vengono svolti da coprocessori - il Denise, il Paula ed il Gary - da un processore grafico, il Copper, e da un velocissimo manipolatore - trasferitore di memoria, il Blitter (gli ultimi due risiedevano in due integrati separati nell'A1000, ed in un unico integrato, il Fat Agnus, nei modelli 500,2000 e 2500).

Il 68000 é il capostipite di una famiglia di processori, nata nel 1978. Infatti esistono altri modelli di processori 68000-compatibili: il 68008, il 68010, il 68020, ed il 68030. Tutti questi hanno piedinatura differente dal 68000 (tranne il 68010), differente velocitá e alcune differenze (molto lievi) nel set di istruzioni (che resta comunque 68000-compatibile, tranne una famosa eccezione - la maledetta MOVE SR,<da/ea> - di cui parleremo in futuro). Qui a fianco vedete la tabella con le differenze tra i vari processori.

C'é una caratteristica nella tabella che non é stata descritta in precedenza: la possibilitá di utilizzare memorie di massa (hard disk) come se fossero comunissime RAM; questa operazione (per la

Bus dati interno,bit	32	32	32	32	32
Bus dati esterno,bit	16	8	16	32	32
Freq. di clock, MHz	7	7	14	14	14
N° registri normali	15	15	15	15	15
N° registri speciali	4	4	7	7	7
N° di interrupts	7	4	7	7	7
Memoria indirizzabile	16 MB	1 MB	16 MB	16 MB	16 MB
Memorie virtuali	No	No	Sì	Sì	Sì
Gestione int. errori	Sì	Sì	Sì	Sì	Sì

quale il processore deve essere predisposto) si svolge tramite un integrato chiamato MMU, Memory Management Unit.

Comunque, il corso di linguaggio macchina che sta per iniziare, descrive anche quelle poche differenze fra i codici macchina degli altri 680xx, per cui sará utile anche per chi possiede una scheda acceleratrice.

E' da notare che la Motorola distribuisce, oltre che ai suddetti processori, anche una serie di coprocessori ausiliari, per specifici compiti: i coprocessori matematici MC68881 e MC68882, la MMU 68851. La loro programmazione non sará descritta in questo articolo, ma sará forse un argomento che tratteremo in futuro, se la diffusione di questi coprocessori aumenterá in maniera considerevole.

Ma ora iniziamo con il corso di linguaggio macchina; specifichiamo subito che alcuni termini non sono descritti all'interno dell'articolo, ma in un glossarietto a parte: questi termini sono indicati da un asterisco (*)

I registri

Unitá fondamentali nell'elaborazione dei dati in un processore sono i registri. Questi si dividono in due categorie: i registri utente (in cui il processore svolge tutte le operazioni) e i registri di sistema (utilizzati dal processore per svolgere compiti interni).

Ma cosa é un registro? Immaginate un allineamento di 32 flip-flop (*): come da numerazione binaria é possibile ottenere da questi 32 flip-flop un numero che varia da 0 a (2^32)-1 = 4294967296 decimale, $FFFFFFFF Hex. In questo registro é possibile appunto memorizzare un numero compreso tra questi due valori, ed é tramite i registri che il processore compie la maggior parte delle operazioni: per esempio, per sommare due numeri A e B, bisogna prima porre il valore A in un registro, poi porre il valore B in un altro registro, quindi sommare il contenuto dei due registri (in uno dei quali verrá posto il risultato, alla fine). Nel 68000 i registri utente si dividono in due gruppi: gli 8 REGISTRI DATI, che sono quelli adibiti alle operazioni utente, ed i 7 REGISTRI INDIRIZZI, adibiti ad operazioni di gestione della memoria. I registri dati sono, come detto sopra, otto e si identificano con i nomi D0,D1,D2,D3,D4,D5,D6 e D7.

Cosí come i registri dati, esistono 7 registri indirizzi, che si identificano con i nomi A0,A1,A2,A3,A4,A5 ed A6 (esiste anche un registro A7, che sará peró descritto tra i registri di sistema). I registri dati possono essere utilizzati indifferentemente a 8, 16 o 32 bit, mentre I REGISTRI INDIRIZZI POSSONO ESSERE

UTILIZZATI SOLO A 32 BIT. Inoltre nei registri indirizzi non possono essere svolte operazioni matematiche (tranne l'addizione e la sottrazione), mentre i registri dati non possono essere utilizzati nella gestione della memoria.

Modo supervisore e modo utente

Il 68000 puó lavorare in due "modi": il primo é il modo supervisore, ed il secondo é il modo utente. Ma cosa vuol dire ??? Praticamente, nel 68000 esistono alcune istruzioni (dette "istruzioni privilegiate") che possono essere eseguite solo in modo supervisore: il loro utilizzo in modo utente provoca un errore detto "violazione di privilegio" (Guru Meditation #0000000008). In un sistema multitasking come l'Amiga, il modo supervisore é utilizzato dall'Exec (*), mentre il modo utente é utilizzato dai vari task (*). In ogni caso é sempre possibile passare dall'uno all'altro stato: si puó definire quindi il modo supervisore uno "stato di emergenza" che il processore utilizza in casi molto particolari (interrupts, errori di programmazione, reset ecc.).

P.S.: l'istruzione MOVE SR,<da/ea>, che costituisce la principale fonte di incompatibilitá fra 68000-68008 e 68010-68020-68030, ha questa particolaritá: sui primi due processori non é un'istruzione privilegiata, mentre gode di questa fastidiosa caratteristica su 68010 e superiori; quindi un programma scritto per 68000 potrebbe causare un'improvvisa violazione di privilegio (da addebitarsi all'istruzione appena vista) su un'altro processore. In ogni caso si tratta di un'istruzione piuttosto rara, quindi il problema si pone in maniera molto esigua.

I registri di sistema

Nel 68000 esistono, oltre ai 15 registri utente, altri 4 registri di sistema (7 nei 68010 e superiori). Eccoli:

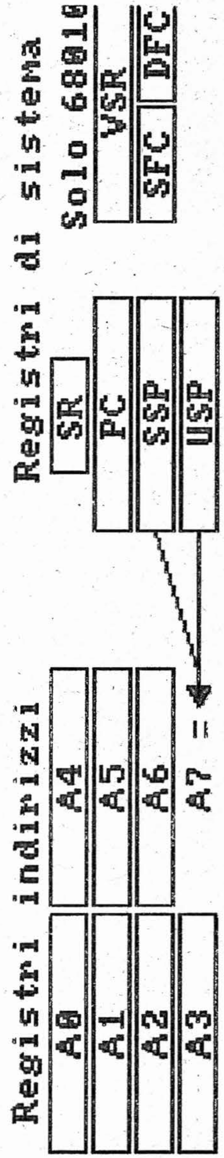
PC = questo registro a 32 bit contiene l'indirizzo della prossima istruzione da eseguire
USP = viene utilizzato come stack pointer (vedere piú avanti) in modo utente
SSP = identico a USP, ma usato in modo supervisore
SR = questo é un registro a 16 bit, utilizzati singolarmente per indicare il verificarsi di alcune condizioni all'interno del processore stesso. Sará descritto in un paragrafo piú avanti
VSR (solo 68010 e superiori) = viene utilizzato per puntare

all'area di memoria contenente gli indirizzi delle routines di eccezione (*) (verrá descritto in seguito quando parleremo delle eccezioni). Nel 68000-68008 non esiste in quanto l'area di memoria dei vettori di eccezione inizia sempre all'indirizzo 0.

SFC e DFC (solo 68010 e superiori) = sono due registri a 3 bit utilizzati per comunicare con un'eventuale MMU 68851 (che il 68000 non puó gestire).

Registri dati

D0	D4
D1	D5
D2	D6
D3	D7



Nelle tabelle vedete lo schema dei registri dell'Amiga.

Gli Stack Pointer USP e SSP

Sono due registri a 32 bit utilizzati rispettivamente in modo utente e in modo supervisore: il loro compito é identico. Puó capitare talvolta che il processore debba memorizzare in RAM un valore che dovrá essere ripescato in seguito (esempio: quando il registro X - che conteneva un importante valore - deve essere utilizzato per altri scopi; il suo contenuto viene salvato in memoria e recuperato al termine delle operazioni): in tal caso USP e SSP contengono l'indirizzo dell'ultimo valore da recuperare.

E' importante notare che se si salvano dei dati sullo stack, essi dovranno essere recuperati nell'ordine inverso a quello in cui sono stati inseriti: se si salvano sullo stack, per esempio, D0, D1 e D2 si dovranno recuperare dallo stack D2, D1 e D0, pena un irrimediabile scambio dei dati: immaginate una pila di

libri: l'ultimo che avete sistemato in cima sará il primo ad essere prelevato, il penultimo sará il secondo e cosí via. Parleremo piú avanti di stack pointers. Per ora basti sapere che questi due registri vengono utilizzati citando A7 o SP (il che é equivalente): si otterrá l'accesso ad USP o SSP a seconda del modo in cui si trova il 68000 in quel momento. Da stato supervisore é peró possibile accedere allo stack pointer utente USP (mentre da stato utente NON é possibile accedere a SSP).

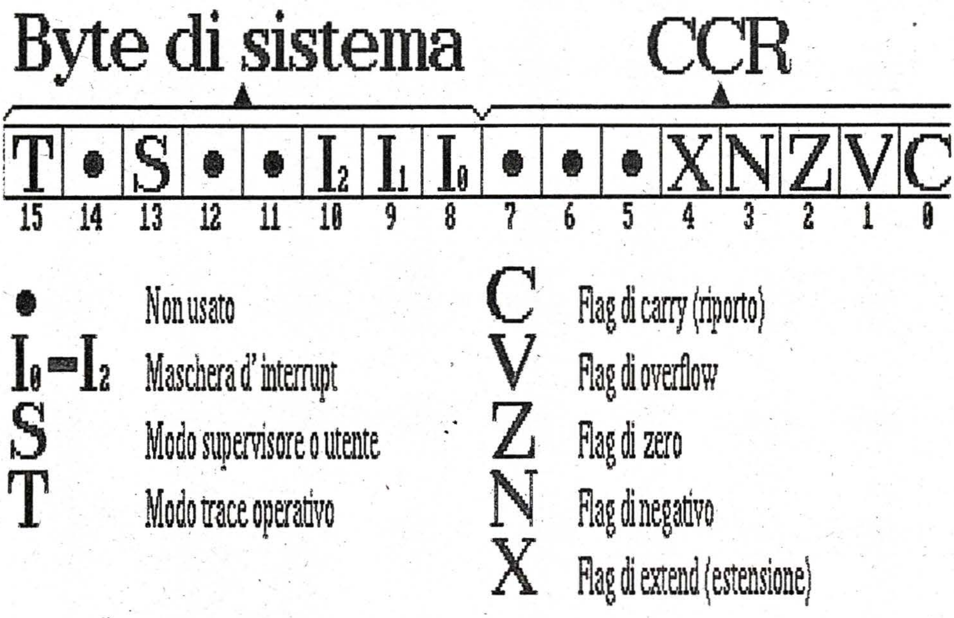
Il registro di stato SR

Questo é un registro a 16 bit; questi bit vengono letti singolarmente per indicare il verificarsi di una condizione nel processore. Ogni accesso in scrittura allo SR é un'istruzione privilegiata, ma é possibile accedere agli 8 bit inferiori dello SR specificando il registro CCR (Condition code register = registro dei codici di condizione). Un accesso in scrittura al CCR NON é istruzione privilegiata. I bit del registro di stato vengono utilizzati quasi tutti singolarmente, e servono ad indicare uno stato del processore o il verificarsi di una condizione. I 16 bit sono:

BIT N.	SIGLA	Nome	Spiegazione
0	**C**	**FLAG DI CARRY**	Se questo bit é settato (posto a 1), significa che si é verificato un "riporto" nell'ultima operazione
1	**V**	**FLAG DI OVERFLOW**	Se questo bit é settato, l'ultima operazione ha generato un risultato che esce fuori dai limiti imposti (byte, word o longword)
2	**Z**	**FLAG DI ZERO**	Se questo bit é posto ad 1 l'ultima operazione ha generato come risultato 0
3	**N**	**FLAG DI NEGATIVO**	Indica, se settato, che l'ultima operazione ha generato un risultato negativo
4	**X**	**FLAG DI EXTEND**	Questo flag é simile al CARRY, ma viene utilizzato in particolari condizioni
5-7	-	**NON UTILIZZATI**	Questi flag possono essere settati ad 1 o a 0, ma non hanno utilizzi
8-10	**I**	**INTERRUPT MASK**	Questi 3 bit (riuniti a formare un numero da 0 a 7) indicano al processore quali interrupt riconoscere e quali no. Nell'Amiga DEVONO ESSERE SETTATI TUTTI A 1 (per rispettare le regole del multitasking)
11-12	-	**NON UTILIZZATI**	Questi flag possono essere settati ad 1 o a 0, ma non hanno utilizzi
13	**S**	**SUPERVISOR FLAG**	Questo flag, se settato, pone il processore in modo supervisore, se resettato in modo utente
14	-	**NON UTILIZZATO**	Questo flag puó essere settato ad 1 o a 0, ma non ha utilizzi
15	**T**	**TRACE MODE FLAG**	Questo flag, se settato, indica che il

processore é posto in modo trace (verrá spiegato in seguito)

Come detto, i bytes 0-7 possono essere indirizzati specificando il registro CCR al posto di SR: ovviamente SR é un registro a soli 16 bit e CCR é un registro a soli 8 bit. Ecco quindi schematizzati in una tabella di riferimento:



Sintassi

Passiamo ora ad un po' di sintassi. Parliamo inanzitutto dei 3 suffissi: .B,.W e .L; come detto in principio, sebbene i registri utente del 68000 siano tutti a 32 bit, i registri dati (E SOLO QUESTI) possono essere utilizzati anche ad 8 e 16 bit. Ma come??? Semplicissimo: dopo l'istruzione da svolgere viene aggiunto il suffisso .B se si tratta di un'operazione ad 8 bit, .W se si tratta di un'operazione a 16 bit e .L se si tratta di un'operazione a 32 bit; se nulla viene specificato, .W é assunto di default (tranne in casi forzati: per esempio operazioni con CCR saranno sempre ad 8 bit!). Se viene utilizzato un movimento a 16 o 8 bit, i 16 o 24 bit superiori rimarranno immutati. Poniamo

alcuni esempi:

MOVE.L D0,D1 trasferisce TUTTI i 32 bit del registro D0 nel registro D1

MOVE.W D1,D2 trasferisce i 16 bit inferiori di D1 nel registro D2, lasciando immutati i bit 16-31 di quest'ultimo

MOVE.B D2,D3 trasferisce gli 8 bit inferiori di D2 nel registro D3, lasciando immutati i bit 9-31 di quest'ultimo

Nota: IN TUTTE LE OPERAZIONI 68000 VIENE SPECIFICATO PRIMA L'OPERANDO SORGENTE E POI IL DESTINATARIO: CIO' VUOL DIRE CHE IL CONTENUTO DEL REGISTRO (O LOCAZIONE DI MEMORIA) SPECIFICATA PER PRIMO VERRA' MOSSO NEL REGISTRO (O LOCAZIONE DI MEMORIA) SPECIFICATA PER SECONDA (Come NON avveniva su altri vecchi processori, come lo Zilog Z80)

Ma l'istruzione MOVE non serve solo a muovere dati da un registro ad un altro, ma anche a muovere dati diretti in un registro: un dato diretto é un numero specificato nel programma, che sintatticamente deve essere preceduto dal simbolo #.

Non é chiaro? In parole povere: prima potevate effettuare solo movimenti da un registro ad un altro, mentre ora é possibile porre un determinato valore (detto dato diretto) in un registro. I movimenti diretti possono essere effettuati sia in formato byte, che in formato word che in formato longword. Quindi:

```
MOVE.L #$FFFFFFFF,D0
MOVE.L #$07654321,A0
MOVE.W #$5000,D1
MOVE.B #$10,D3
```

sono espressioni corrette, ma

```
MOVE.W #$8000,A1
MOVE.W #$10000,D1
MOVE.L D1,#$ABCDEF01
```

sono tre espressioni sbagliate: da quanto abbiamo detto prima dovreste essere capaci di capire il perché. Se proprio non potete fare a meno di ragionare con la vostra testa, continuate a leggere.

Le risposte sono:

1) A1 é un registro indirizzi, che non puó essere utilizzato se non

a 32 bit;

2) $10000 é un numero a 17 bit, che quindi non rientra nel range di una word;

3) Come fate a porre un registro in un numero? Sono stati scambiati operando sorgente e destinazione

Utilizzo della memoria

Ovviamente il microprocessore deve lavorare anche in collaborazione con la memoria. Un primo contatto memoria->processore avviene durante la lettura del programma (che, guarda caso, risiede in memoria - ROM (*) o RAM (*) che sia), ma é anche possibile spostare dati dai registri alla memoria, dalla memoria ai registri o porre un valore diretto in memoria.

Ma come viene gestita la memoria? Esistono varie locazioni di memoria, tutte identificate da un indirizzo che va via via crescendo; ciascuna di queste locazioni puó contenere un valore ad 8 bit (da 0 a 255). Inoltre le locazioni RAM possono essere lette e scritte, mentre le locazioni ROM possono essere solo lette: un tentativo di scrittura in esse non provocherá crash del sistema ma non avrá effetto.

Troppo difficile? Ricorriamo ad un esempio: immaginate una serie di scatole numerate progressivamente, ciascuna contenente un certo numero di biglie. Per identificare una scatola, voi utilizzate il suo numero e poi, una volta riconosciutala, potrete controllare o variare il contenuto a vostro piacere. Esistono peró due tipi di scatole: alcune sono aperte e potete sia guardare che modificare il contenuto (RAM), altre sono coperte da un vetro e potete solo guardare (ROM).

Cosí come per le operazioni tra i registri, anche per le operazioni con la memoria sono possibili indirizzamenti a modo byte, a modo word e a modo longword. Qui peró la cosa funziona in modo differente: infatti, se un'operazione di formato byte riguarda solo la cella di memoria interessata, un'operazione di tipo word riguarda la locazione specificata e la successiva, e un'operazione di range longword riguarda ben 3 locazioni di memoria, oltre a quella specificata. Quindi le operazioni di tipo:

```
MOVE.B #$AB,$7e800          Porrá il valore $AB nella locazione $7e800
MOVE.W #$ABCD,$7e800        Porrá $AB nella locazione $7e800 e $CD
                                                nella locazione $7e801
MOVE.L #$ABCDE01,$7e800 Porrá $AB in $7e800, $CD in $7e801, $DE
```

in $7e802 e $01 in $7e803

Similmentè

MOVE.L D0,$7e800 porrá i bit 31-24 di D0 in $7e800, i bit 23-16 in $7e801, i bit 15-8 in $7e802 e i bit 7-0 in $7e803.

Notare la sintassi: la locazione di memoria viene indicata come il valore diretto, ma senza il simbolo '#'. Inoltre, se un'operazione del tipo:

MOVE.B D0,#$7e800 é palesemente scorretta, l'operazione
MOVE.B D0,$7e800 é corretta ed utilizzatissima, e significa "muovi gli 8 bit inferiori di D0 nella locazione di memoria $7e800.

Cosí come nei movimenti da registro (o valore diretto) a memoria, anche nei movimenti da memoria a registro di lunghezza superiore al byte vengono interessate piú locazioni, la prima delle quali conterrá i bit piú alti e l'ultima i bit piú bassi (ossia: in caso di movimento word, la prima locazione conterrá i bit 15-8 e la seconda i bit 7-0, ed in caso di movimento di lunghezza longword, la prima locazione conterrá i bit 31-24, la seconda i bit 23-16, la terza i bit 15-8 e la quarta i bit 7-0; questi bit verranno trasferiti nell'ordine appena detto all'interno del registro) come NON AVVENIVA NELLO Z80.

Due limitazioni: TUTTI GLI INDIRIZZAMENTI DI TIPO WORD E LONGWORD DEVONO INIZIARE AD UN INDIRIZZO PARI, PENA UN "ADDRESS ERROR" (Guru Meditation #00000003) e LA MEMORIA INDIRIZZABILE DAL 68000 PARTE DA $00000000 E TERMINA A $00FFFFFF = 16MB (quella del 68008 - e solo questo - termina a $000FFFFF = 1MB)

I registri indirizzi e la memoria

Abbiamo detto che i registri indirizzi servono per operazioni con la memoria: ma come? E' semplicissimo. Negli esempi visti sopra, la locazione di memoria in causa era fissa, e decisa dal programma; con lo svantaggio che se un'improvvisa carenza di memoria facesse cambiare l'indirizzo di quella variabile, il programma non funzionerebbe piú. Quindi é possibile utilizzare i registri indirizzi: il numero contenuto nel registro indirizzi in causa indica quale locazione di memoria utilizzare. La sintassi prevede il numero del registro indirizzi scritto tra parentesi; quindi:

MOVE.L #$4,A0
MOVE.L (A0),D0 é equivalente a MOVE.L $4,D0

Da questo punto di vista non si vedono immediati vantaggi: ma se nel programma la locazione in uso dovesse cambiare, sará sufficiente cambiare (sempre all'interno del programma) il contenuto del registro A0, senza dover riscrivere il programma punto e daccapo. Inoltre i registri indirizzi consentono alcune facilitazioni, dal punto di vista dell'utilizzo della memoria. Ma analizzeremo tutto nel prossimo numero, anche perchê penso che abbiamo giá discusso abbastanza su un argomento piuttosto ostico di per sē.

SE NON AVETE COMPRESO QUALCOSA, RILEGGETE CON CALMA LA PARTE PRECEDENTE AL PARAGRAFO NON CHIARO E SOPRATTUTTO CERCATE DI ARRIVARE AL PROSSIMO NUMERO CON LE IDEE BEN CHIARE.

Sul prossimo numero tratteremo:

1) Le istruzioni ADD e SUB, somma e sottrazione
2) L'istruzione DBF, per creare dei cicli interni al programma
3) Nuovi modi per indirizzare la memoria
4) Come leggere le Guru Meditation

Glossario

FLIP-FLOP : dispositivo logico in grado di conservare un bit (acceso o spento) fino al successivo reset o all'interrompersi dell'alimentazione.

EXEC : libreria di Amiga atta a svolgere compiti di basilare importanza, come la gestione di più programmi in funzione contemporaneamente, il processo di reset, l'allocazione della memoria, le Guru Meditation, ecc.

TASK : un programma che funziona tramite l'Exec, che provvederà a far funzionare più di un task con un solo processore, se necessario.

ECCEZIONE : evento tramite il quale una periferica esterna o un errore interno della CPU la costringono ad interrompere il programma attualmente in esecuzione e ad eseguire un programma atto a gestire l'evento verificatosi, programma che è detto "routine di eccezione"

RAM : Random Access Memory (Memoria ad accesso casuale) : quella parte della memoria del computer atta ad essere modificata, ma che perde il suo contenuto allo spegnimento della macchina.

ROM : Read-Only Memory (Memoria a sola lettura) : tipo di memoria che viene scritta una sola volta, alla sua creazione, ed il cui contenuto non può essere modificato e non viene perso allo spegnimento della macchina (solitamente, in un computer contiene il sistema operativo).

DUNGEON MASTER

Dungeon Master appartiene a quella categoria di giochi tratti dall'ormai diffusissimo gioco "di ruolo" che è Dungeons & Dragons.

Così come i suoi consimili Phantasie I e III, e la famosissima saga di Bard's Tale questo gioco narra le vicissitudini di un gruppo di personaggi in un mondo fantastico dove operano la magia e l'arte del combattimento.

Scopo di Dungeon Master è quello di recuperare una spada magica nascosta nei recessi di un labirinto, popolato dalle più svariate e crudeli creature. Ma più si avanza in questo labirinto, più forti e temibili diventano i mostri da affrontare, più sottili gli enigmi da sciogliere per poter proseguire e più potenti diventano gli incantesimi utilizzabili da personaggi e mostri.

I personaggi che compongono il gruppo possono essere scelti dal giocatore all'inizio della partita (che può essere in qualsiasi momento interrotta e salvata su disco) ed hanno caratteristiche che li rendono molto simili agli esseri umani "veri" quali la necessità di mangiare, bere e riposarsi ed hanno la particolarità di progredire nelle quattro arti principali (combattimento con armi, arti marziali - inclusa l'abilità di lanciare oggetti ed armi, arti magiche offensive - per maghi - e difensive e di guarigione - per chierici) con il costante allenamento. Inoltre ogni personaggio (ne esistono circa una trentina, dei quali ne possono essere scelti quattro, che non potranno più essere cambiati) è più portato verso alcune delle quattro arti e meno verso altre.

All'inizio ciascun personaggio ha in dotazione una certa quantità di oggetti personali che potrà essere notevolmente accresciuta da una costante ed minuziosa ricerca all interno del labirinto (che d'ora in poi chiameremo con il suo vero nome: Dungeon).

Ma perché Dungeon Master ha avuto questo eccezionale successo? Quale è il segreto dell'ultimo gioco della FTL?

La prima cosa da tener presente è che non è stato tralasciato nessun particolare nella stesura del programma, tanto è vero che gira solo su macchine dotate di almeno un megabyte di RAM. Gli oggetti disponibili sono più di un centinaio, ciascuno dotato delle proprie caratteristiche (magiche e non); le ventiquattro magie, ciascuna delle quali può essere pronunciata in sei modi diversi, corrispondenti a sei diversi livelli di potenza. Inoltre, per ogni personaggio il programma mantiene costantemente sotto controllo diversi dati riguardanti il carico

e quindi la velocità di spostamento, la stanchezza e la resistenza, la fame e la sete.

Oltre a questo il programma gestisce anche la posizione degli oggetti raccolti (in tasca, nello zaino, in mano) e delle armature (che sono divise in elmi, corazze, scudi, calzature e ginocchiere).

Secondo particolare, i mostri. Sebbene il loro numero non sia molto elevato, sono disegnati ed animati con particolare cura; inoltre ogni mostro ha le sue caratteristiche ed un suo particolare comportamento, gestito con eccezionale fedeltà.

Terzo, il labirinto, che non si presenta come il "solito" labirinto 3D per computer, ma è disegnato con tale cura e maestria da sembrare quasi vero. Scritte sui muri e pulsanti nascosti nella roccia, così come pozze d'acqua, fontane, sculture e crepe nella roccia, disegnate a regola d'arte, sono caratteritiche proprie di questo programma.

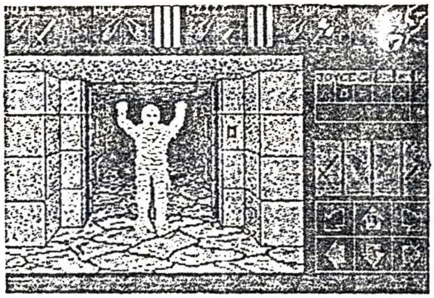
Inoltre l'uso del programma è quanto mai intuitivo, in quanto l'utente viene "iniziato" a particolari quali i pulsanti nel muro o ai trabocchetti tramite alcune pergamene, che risultano frequentissime ai livelli inferiori ed il cui numero va sempre più ad affievolirsi nei livelli inferiori.

L'audio (stereo: se un mostro si avvicina da sinistra, per esempio, sarà attraverso l'altoparlante sinistro del monitor che si udiranno i suoi passi) , l'eccezionale realismo dei combattimenti e del lancio delle magie e la fedeltà all'originale gioco da tavolo Dungeons & Dragons, sono altri particolari non sottovalutabili nel giudizio di questo programma.

Insomma, se siete amanti dei giochi di ruolo Dungeon Master non deve assolutamente mancare alla vostra softeca (in versione originale, perché il manuale è indispensabile e le copie reperibili sul mercato pirata sono estremamente difettose).

Se invece siete il tipo da shoot-em-up, Dungeon Master non fâ per voi.

Un voto ? In una scala da uno a dieci, Dungeon Master vale undici: provare per credere.



DUNGEON MASTER HELP

Dungeon Master non è certamente un gioco facilmente risolvibile, anzi presenta notevoli difficoltà, specialmente per quanto riguarda gli incantesimi ed i passaggi di livello nelle varie arti. Allo scopo di agevolare i giocatori che non hanno la "pazienza" di risolvere tutto il gioco con le loro forze, (impresa che richiede molti giorni, se non mesi di impegno) pubblichiamo una serie di consigli atti a favorire il giocatore nell'uso dell'programma, e nella sua comprensione.

1) Operare una attenta riflessione nella scelta iniziale dei personaggi, meglio perdere un poco di tempo ad esaminare buona parte dei personaggi all'inizio del gioco, che trovarsi in serie difficoltà più avanti a causa delle carenze dei personaggi scelti.

2) Ricordarsi di far esercitare spesso i personaggi nell'uso degli incantesimi, questa pratica è essenziale nei primi livelli del dungeon, a causa della scarsa familiaretà dei personaggi con essi.

3) Cominciare sempre pronunciando incantesimi di basso potere, per poi passare, con l'aumento dell'esperienza dei personaggi, ad incantesimi con un potere maggiore.

4) Esercitarsi spesso nel lancio di oggetti e nel combattimento senza armi: è l'unico modo per progredire nelle arti marziali.

5) Crearsi sempre una vasta scorta di pozioni, (sopratutto quelle anti-veleno e quelle curative per ripristinare i punti-ferita) per non dover trovarsi durante un combattimento con la necessità di crearsene.

6) Salvare spesso la partita in corso su disco, questa operazione, oltretutto molto semplice e veloce, vi risparmierà la seccatura di veder sprecato tutto il vostro lavoro per una scelta sbagliata.

7) Nella scelta iniziale, preferire la creazione di gruppi compositi, cercando però di evitare personaggi (come Hulk il barbaro), "congenitamente" impossibilitati a progredire nelle arti magiche in quanto dotati di zero punti magici (non possono lanciare incantesimi!!). Un gruppo che possiamo consigliarvi per la sua efficienza è quello composto nell'ordine da: Daroou, che pur essendo un fortissimo guerriero(tra l'altro è dotato di un altissimo numero di punti ferita) può divenire, seppur faticosamente, altrettanto efficiente nelle varie arti; Zed il duca, che è certamente

il personaggio più completo a disposizione (è inoltre l'unico personaggio dotato fin dall'inizio di una armatura metallica completa); Mophus il guaritore, un eccellente chierico e infine Gothmog certamente il miglior mago a disposizione.

8) Per risparmiare ai giocatori l'improba fatica di ricercare le ventiquattro pergamene con le altrettante magie(molto delle quali sepolte molto in profondità), pubblichiamo in calce, la lista completa delle stesse.

P.S. Tenendo presente che ogni magia è composta da due a quattro parole magiche Eviteremo di riportare tutte le quattro serie di sei parole magiche, che combinandosi danno vita agli incantesimi, sostituendole con tre serie di numeri da uno a sei, poiché la prima serie di parole che indicheremo con X è fissa per tutti gli incantesimi , e rapresenta i sei possibili livelli di potenza degli stessi (per esempio X.4.4 sta per 1.4.4 = palla di fuoco di bassa potenza, 2.4.4 = palla di fuoco di medio-bassa potenza, ... , 6.4.4 = palla di fuoco extra extra strong).

X.1= Pozione per riacquistare la resistenza
X.2= Pozione cura ferite
X.4= Luce magica
X.6= Apertura e chiusura porte (non tutte : anzi quasi nessuna)
X.1.4= Schermo collettivo anti-magia
X.1.5= Schermo personale anti-magia
X.2.5= Pozione contro veleno
X.3.1= Nube di veleno
X.4.4= Palla di fuoco
X.5.2= Disperde gli esseri immateriali (fantasmi & Co.)
X.6.1= Riempie di veleno una fiasca per lanciarla (ma perderete la fiasca)
X.1.5.2= Crea una serie di impronte di riferimento lì dove il gruppo è passato
X.1.5.3= Pozione per aumentare temporaneamente la saggezza
X.1.5.4= Pozione per aumentare temporaneamente la vitalità
X.3.2.5= Consente di guardare attraverso i muri - X indica la DURATA
X.3.2.6= Invisibilità
X.3.3.5= Missile magico
X.3.4.5= Luce persistente
X.3.5.2= Pozione per aumentare temporaneamente la destrezza
X.4.5.1= Pozione per aumentare temporaneamente la forza

X.4.5.4= Resistenza al fuoco
X.5.4.6= Oscurità
X.6.3.5= Questo incantesimo crea una pietra magica (servirà al livello #13)
X.6.5.5= Pozione per aumentare i punti magici

9) Quando vi riposate, seguite questa procedura : togliete di mano la torcia, se ne avete, e dormite fino a far risalire al massimo i punti magici di tutti i membri del gruppo. Quindi caricatevi le magie che supponete di dover utilizzare e dormite di nuovo, fino al raggiungimento del MASSIMO in punti ferita, resistenza e punti magici : così facendo avrete i punti magici al massimo ed un incantesimo già caricato. Se, mentre dormite, notate che qualcuna delle colonnine scende (ma non di poco) fermatevi e controllate che i vostri personaggi non siano affamati, assetati, avvelenati o troppo carichi.

10) Ricordate che se caricate troppo un personaggio, questi perderà più resistenza degli altri e avrà, rispetto a questi, bisogno di più cibo e acqua.

11) Non fate economia con l'acqua, quando ne trovate. Tanto l'acqua delle fontane non finisce mai.

Eccoci finalmente alla seconda tappa del nostro viaggio alla scoperta dell'AMIGA.
Come promesso, questo mese esamineremo due delle principali caratteristiche di Amiga: Il MultiTasking e L'Intuition.

Il MultiTasking

La parola MultiTasking vuol dire letteralmente lavoro multiplo; in sostanza,lavorare in MultiTasking vuol dire avere la possibilità di far girare più di un programma contemporaneamente il che è una vera benedizione, ad esempio, per chi programma.
Immaginate infatti di stare scrivendo un programma e di avere bisogno di dare un'occhiata alla routine di un'altro programma; in un sistema non dotato di MultiTasking dovreste resettare o perlomeno eliminare dal video il vostro lavoro corrente per visualizzare l'altro programma.

Amiga Da 0 - Parte II

Su Amiga invece basterà aprire un'altra finestra di lavoro, ad esempio un'altro CLI, e caricare i dati richiesti.

Comunque è assolutamente sbagliato associare l'idea di Task con quello di finestra o comunque con qualcosa di visibile; infatti, anche quando l'utente non esegue alcuna operazione, l'Amiga mantiene comunque dei task aperti, basti pensare a tutte le librerie, i programmi di controllo dei dischi fisici e virtuali, il Workbench.

Naturalmente l'uso dei Task richiede memoria, quindi più ce n'è più se ne possono aprire, a discapito però della velocità che si riduce notevolmente se alcuni Task utilizzano contemporaneamente lo stesso coprocessore, ad esempio per la generazione di grafica per operazioni matematiche particolarmente complesse ecc.

Il multitasking può però risultare piuttosto svantaggioso nel caso in cui due task utilizzino contemporaneamente lo STESSO DISCO (unità a floppy o a dischi fissi che sia) : infatti, in questo caso, la testina del dispositivo è costretta a dondolare fra la posizione sul disco indicata dal primo task e quella indicata dal secondo; questo movimento supplementare rallenta di circa il 50% la velocità di accesso al disco. L'accesso di due task a due DISCHI DIVERSI non comporta invece rallentamenti considerevoli.

L'Intuition

Qual'è la cosa che vi ha colpito di più quando avete visto per la prima volta l'Amiga oltre alla grafica e al suono?

Le icone (quei quadratini a forma di dischetto che appaiono sullo schermo del Workbench)? I menu a discesa? I Gadget (quei piccoli "oggetti" disegnati sul video che vi permettono, una volta "afferratili" con il puntatore del mouse, di posizionere le finestre una dietro l'altra o di ridimensionarle)?

Tutti particolari, quelli citati sopra, che consentono un facilissimo contatto fra l'utente e la macchina, contatto che avviene tramite un puntatore sul video controllato dal mouse. La cosa interessante è che gadget, menu a discesa e altri particolari del genere sono comuni a tutti i programmi, ed il loro utilizzo, quanto mai semplice, evita il dover passare per la tastiera.

Ma cosa c'entra tutto questo con Intuition?

Bene, tutto questo è frutto di Intuition. Tecnicamente Intuition è una libreria che occupa ben 80K dei 256 del Kickstart, le cui routines servono appunto alla gestione di quegli "oggetti" che rendono quanto mai intuitivo (da qui il nome) il contatto fra l'utente e la macchina.

Ma ora vediamo come

funzionano praticamente i menu a discesa i gadget e simili.

Come avrete notato il mouse di Amiga ha 2 tasti con funzioni separate; quello di sinistra ha la funzione di selettore per i gadget delle finestre per le icone e per il posizionamento delle finestre e dei menu; quello di destra, invece, serve per attivare i menu a discesa.

Le finestre

Componenti fondamentali di Intuition sono le finestre.

Queste ultime sono costituite da una porzione rettangolare di video contenente testi o immagini. Ma la finestra non è solo una semplice porzione di video. Infatti la semplice "fetta" di video è detta LAYER, ma le finestre sono sempre dotate di aggiunte supplementari: un'intestazione (detta titlebar) è SEMPRE presente, anche se a volte è coperta da altri layers; spesso la finestra è contornata da un bordo (le finestre senza bordi sono dette BORDERLESS WINDOWS), possono essere presenti 4 gadgets piccoli ed uno largo quasi quanto l'intera finestra detto DRAG BAR.

Portando il puntatore del mouse su questo gadget (che, se presente, copre gran parte del lato superiore della finestra), premendo il bottone sinistro del mouse, mantenendolo premuto e spostando il mouse, si sposterà anche la finestra per quanto le dimensioni del video lo consentono.

La presenza della drag bar è segnalata da due righe orizzontali dopo il titolo della finestra (ma a volte il titolo è così lungo da uscirne fuori e da non permettere la visualizzazione delle strisce orizzontali): tuttavia, se presente, la drag bar copre ANCHE IL TITOLO DELLA FINESTRA (il che vuol dire che potete afferrare la drag bar anche clickando sul titolo).

La drag bar (e il titolo) hanno anche un'altra funzione: quella di segnalare se la finestra è attiva o no (ma di questo parleremo più avanti).

Passiamo quindi ai due gadgets generalmente presenti nell'angolo superiore destro della finestra: i DEPTH GADGETS, gadget di profondità.

Così come fogli di carta sovrapposti sparsi su un tavolo, anche le finestre possono essere poste l'una sull'altra e scambiate di profondità: ossia la finestra che stava "sotto" può essere posta "sopra" alle altre. Queste operazioni possono essere compiute portando il puntatore del mouse su uno dei due gadget posti nell'angolo in altro a destra della finestra (su cui sono disegnati due rettangoli, uno nero ed uno bianco, che si sovrappongono) e premendo il pulsante sinistro del mouse

(non è necessario mantenerlo premuto): per essere più precisi il gadget di sinistra spinge la finestra interessata DIETRO alle altre (infatti più di due finestre possono essere sovrapposte), mentre il gadget di destra porta la finestra interessata AVANTI alle altre.

Esistono però alcune finestre, dette BACKDROP WINDOWS, che compaiono sempre dietro a tutte le finestre e che non possono mai essere portate di fronte ad altre: un esempio di queste è la finestra del workbench, che, anche se completamente invisibile (la sua title bar è nascosta dall'intestazione "WORKBENCH RELEASE 1.x" ed è una finestra BORDERLESS), esiste (è su questa che vengono disegnate alcune icone, come quella del RAM DISK), ma non può essere portata avanti a nessuna finestra esistente sul video. In ogni caso le BACKDROP WINDOWS non possono mai essere sovrapposte.

Veniamo al gadget posto in basso a destra in una finestra, il SIZING GADGET.

Quest'ultimo serve a cambiare le dimensioni di una finestra; il suo utilizzo è simile a quello della DRAG BAR: infatti deve essere "afferrato" con il pulsante sinistro del mouse, pulsante che deve rimanere premuto durante tutta la durata dell'operazione e rilasciata al termine. Notate che QUANDO UN GADGET RIMANE PREMUTO, L'INTERO SISTEMA RIMANE BLOCCATO FINO AL TERMINE DELL'OPERAZIONE.

Siamo quindi all'ultimo dei gadget "di sistema" (ossia quelli standard) attaccabili ad una finestra : il CLOSE GADGET.

Quest'ultimo (non presente sulla finestra "Amiga DOS" che compare allo startup della macchina) si trova in alto a sinistra nella finestra e serve a chiuderla ed a segnalare il termine delle operazioni relative a quella finestra (spesso serve anche a chiudere un programma). Per attivarlo portare il puntatore del mouse sul CLOSE GADGET e premere il pulsante sinistro del mouse (d'ora in poi, tranne dove diversamente specificato, "premere" il pulsante sinistro del mouse significa "premere e rilasciare" il pulsante). Salvo crashes del sistema, la finestra o il programma dovrebbe chiudersi.

Finestre "Attive" e "Inattive"

Abbiamo parlato del Multitasking: ma supponiamo di avere due task in esecuzione, ciascuno dotato della propria finestra: come scegliere di inviare dati all'uno o all'altro task? Semplice, ATTIVANDO LA SUA FINESTRA. Normalmente, in Amiga, una sola finestra è attiva in un momento (se ne sono attive due, vi garantisco il crash immediato del sistema in

Amiga Da 0 - Parte II

meno di 500 millisecondi): il che vuol dire che tutti i dati inviati col mouse o con la tastiera vengono inviati SOLO AL PROGRAMMA CUI APPARTIENE LA FINESTRA ATTIVA.

Una finestra viene attivata ponendo il puntatore del mouse nella sua "zona video", e premendo il pulsante sinistro del mouse.

Si può distinguere lo stato di una finestra osservando la sua titlebar: se è disegnata normalmente, la finestra è attiva; se è disegnata come se fosse coperta da una rete (tecnicamente si dice "ghosted") la finestra non è attiva.

Ribadisco che tutti gli eventi che si verificano vengono inviati solo al programma cui appartiene la finestra attiva in quel momento : quindi, fate attenzione a dove scrivete i vostri dati !!!

ECCEZIONE : l'inserimento o l'estrazione di un disco e l'attivazione o la disabilitazione di una finestra vengono segnalate anche a finestra inattiva.

Comunque, le caratteristiche di cui abbiamo parlato riguardano anche gli schermi, in quanto anche questi ultimi sono provvisti di DRAG BAR di DEPTH GADGET (ma non di CLOSE e SIZING GADGET) e possono essere attivi uno alla volta, naturalmente basta selezionare una finestra appartenente ad uno schermo perché questo si attivi.

Come è ovvio, l'Intuition non è solo questo in quanto esistono molti altri tipi di gadget (quali i gadget stringa, quelli a mutua esclusione, quelli proporzionali ecc.), ma di questo parleremo la prossima volta.

La nostra avventura inizia in un sistema solare lontano duecento milioni di anni luce; nelle vesti di DANCIS FRAKE dobbiamo recuperare il CRISTALLO DI KRONOS, custodito in un luogo segreto.

Questa in sintesi la trama del gioco dell'ADDICTIVE GAME di strategia, avventura, azione.

Il gioco inizia sul pianeta MELTOCA, nel NOVALA PARK: DANCIS FRAKE si trova lontano un milione di DRIKS dalla sua base, ZAPMINOLA, ma non ricorda cosa sia successo la notte precedente.

Da questo momento saremo noi a controllare, tramite il joystick, il nostro eroe. Incontreremo molti personaggi e dovremo prestare attenzione a ció che diremo o faremo.

Quando incontriamo un personaggio, usiamo il "fire" finché questo non finisce di parlare e se vogliamo dialogare, ci serviremo del tasto funzione "F1". Questo tasto abilita il "TEXT MODE" permettendo di usare la tastiera (compreso il backspace per la correzione di eventuali errori), mentre con "ESC" ritorniamo al modo normale. In genere, i personaggi, rispondono generosamente (e ironicamente) alle nostre domande, ma, occasionalmente, potrebbero rivolgere una domanda e, quindi, esigere una risposta.

Ricordate che non tutti i personaggi sono cosí gentili; spesso dovremo combattere con qualcuno di essi e in questo caso useremo la spada, servendoci del tasto funzione "F9". In questi scontri l'uso del joystick ê abbastanza semplice: gli otto movimenti variano se usiamo o meno il "fuoco".

Sul percorso sono disseminati degli oggetti che vedremo lampeggiare a sinistra del BOX che mostra l'INVENTORY e il punteggio: usiamo "F1" per attivare l'inventario ed "F2" per raccoglierli. Se un personaggio ci dona qualcosa, questa automaticamente passa nell'inventario. Per una descrizione degli oggetti, dobbiamo attivare il display dell'inventario e posizionare, tramite i tasti "F6" ed "F7", l'oggetto nel BOX USE-EXAMINE e poi usiamo "F5". Lo stesso discorso vale per poter usare un oggetto, solo che al posto del tasto F5, dobbiamo usare F10.

Una nota importante:tutti i menú relativi ai tasti funzione, non possono essere attivati se sono presenti le vignette di discorso.

Durante il gioco, possiamo usare la moneta corrente del sistema KREEMA chiamata "SKRINGLES". Questa puó essere usata per ottenere informazioni e oggetti dai vari personaggi, nonché per fare beneficenza. Per usarla,

abilitate il text mode e scrivete l'ammontare della cifra. DANCIS FRAKE puó spostarsi tra i vari pianeti e ingaggiare battaglie spaziali con la sua fantasiosa astronave: scegliete la destinazione e decollate.

Tenendo contemporaneamente il joystick in posizione alto e il fuoco premuto si accelera; si decelera in modo contrario. Il pannello di controllo dell'astronave ê abbastanza semplice da usare: nella colonna a sinistra leggiamo la distanza dalla meta, mentre quella di destra indica le armi. Il quadrante sinistro avverte dell'approssimarsi della meta (quando lampeggia) e il quadrante destro indica la velocitá.

Per salvare il gioco dobbiamo usare il tasto 8 del KEYPAD NUMERICO e seguire le istruzioni sullo schermo.

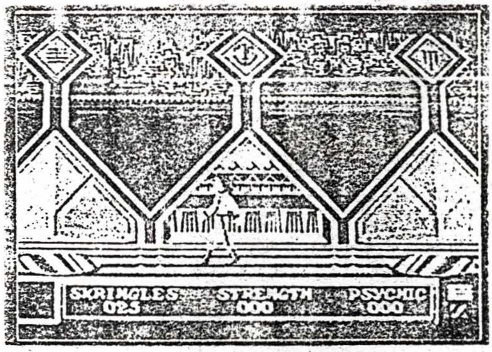
Quando il gioco ê stato salvato, inseriamo subito il disco che era presente in DF0.

Non bisogna assolutamente effettuare la procedura di salvataggio durante un volo spaziale o quando il programma richiede un cambio di disco, pena un CRASH del sistema. Una volta che il gioco ê in corso, si puó caricare la situazione salvata in qualunque momento, anche quando il programma richiede un cambiamento di disco; basta usare il tasto "2" del KEYPAD NUMERICO e seguire le istruzioni. Una volta che i dati sono stati caricati, bisogna inserire il disco richiesto e usare il "FIRE".

Non bisogna mai cambiare un disco a meno che il programma non lo richieda.

Qualche consiglio utile: in base alle azioni che svolgiamo durante il gioco, incrementiamo la potenza fisica (STRENGTH), molto utile nei combattimenti, e la forza psichica (PSYCHIC). Queste due componenti di DANCIS sono molto importanti per giungere alla fine, cosa non molto facile.....

Non dovete peró disperare: noi speriamo di pubblicare la soluzione sul prossimo numero, ma se non ci riusciremo, contiamo sul vostro aiuto. Buona fortuna!



ReportMouse()

La funzione ReportMouse di intuition.library é vittima di un curioso bug: infatti, sebbene la sua corretta dicitura sia:

```
ReportMouse(Window,Boolean)(A0/D0)
```

per un'errore negli stubs dei Lattice C, la funzione é definita come:

```
ReportMouse(Boolean,Window)(D0/A0)
```

Invece negli Aztec C, la funzione é definita correttamente. Onde evitare disguidi, utilizzate sempre la seguente funzione:

```
NewReportMouse(Window,Boolean)
struct Window *Window ;
int Boolean ;
 {
#ifdef LATTICE
  ReportMouse(Boolean,Window) ;
#else
  ReportMouse(Window,Boolean) ;
#endif
 }
```

Errata on Intuition Reference Manual

Anche Robert J. Mical (l'autore di intuition) commette degli errori: infatti nell'Intuition Reference Manual sono presenti alcuni errori (oltre a moltissimi "this function is not implemented now" - questa funzione non é ora implementata - !SIGH!) che possono rendere la vita dura ad alcuni programmatori. Elenchiamo i tre principali errori del manuale:

PAG. 40 - E' stato scritto INTERLACE al posto di LACE (e non solo in questa pagina: l'errore si ripete)

PAG. 134 - Nella definizione della struttura MenuItem é stata omessa una dichiarazione; infatti il manuale riporta:

```
...
LONG MutualExclude
APTR ItemFill
BYTE Command
...
```

mentre la dichiarazione corretta é:

```
...
LONG MutualExclude
APTR ItemFill
APTR SelectFill
BYTE Command
...
```

In ogni caso, la dichiarazione negli includes dell'appendice B é esatta.

PAG 188 e seg. - E' stato scritto XOR al posto di COMPLEMENT

Se proprio volete evitare di confondere le macro dettate da RJ con quelle corrette, inserite all'inizio del vostro programma le linee:

```
#include "intuition/intuition.h" #define INTERLACE LACE
#define XOR COMPLEMENT
```

Per l'errore nella struttura MenuItem, prendete una matita e correggete la pagina 188 del manuale...

ColdReset()

Nell'Amiga ROM Kernel manual si cita una funzione dell'exec.library che é invece inesistente: ColdReset(). E' possibile scrivere l'equivalente di questa funzione con poche righe di codice macchina (versione per Lattice Assembler):

```
        section "ColdReset",CODE
        xref _ColdReset

_ColdReset:
        move.l $4,A6
        addq #1,46(A6)
        move.l #$FC0000,$80
        trap #$0
        END
```

Editare il file con il suffisso .asm e assemblare con ASM FILE.ASM. Otterrete il file FILE.O che dovrete linkare come libreria in tutti i programmi che fanno uso di questa funzione.

I bugs del Lattice C v5.0

Il Lattice C v5.0 é forse il compilatore C che dovrebbe entrare nel Guinness dei Primati per l'eccezionale numero di bugs presenti (si spera bene per la versione 5.02). Citiamo i piú importanti:

1) Le funzioni printf(), sprintf() e fprintf() riconoscono un solo parametro correttamente. Quindi, se dovete utilizzare una di queste funzioni e specificare piú di un parametro, spezzate la stringa in due o piú parti e chiamate piú volte la funzione rea

2) Talvolta il linker si "mangia" il simbolo _stdout dal modulo di startup c.o. NON TENTATE DI REINSERIRLO DA LINKER, CON L'OPZIONE VERBOSE. Quindi, se doveste averne bisogno, aggiungete le seguenti linee:

(subito dopo gli #include, fuori da ogni funzione) int stdout ;
(come prima linea della funzione main())   stdout = Output() ;

Cosí facendo dovrete peró rinunciare alla redirezione dell'output con il simbolo ">" nella linea di comandi: pazienza, prendetevela con John Toebes...

3) Alcuni includes relativi alla libreria exec sono errati. Ricopiateli dal vecchio Lattice C.

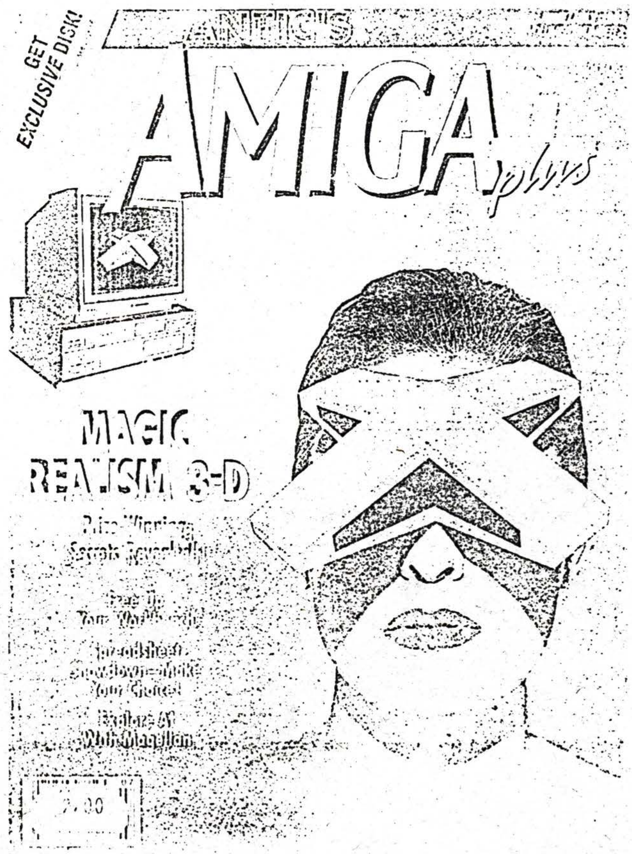
4) Le definizioni di variabili chip e fast non funzionano quasi mai (il compilatore e il linker non generano errori, ma un file non eseguibile...)

DiskFontBase e DiskfontBase (notare la minuscola)

Normalmente, in Amiga, i simboli (tranne quelli delle funzioni della libreria di compilatore) si scrivono con la prima lettera maiuscola. Se il simbolo é composto da piú parole o da sigle la prima lettera di ciascuna parola (o sigla o abbreviazione) é maiuscola (GfxBase, IntuitionBase, MakeVPort - che sta per MakeViewPort).

Eccezione é DiskfontBase, in cui la parola "font" viene scritta con la minuscola: la cosa curiosa é che, se utilizzate DiskFontBase al suo posto, il linker dovrebbe segnalarvi la mancata referenza di uno dei due simboli: IL CHE PUNTUALMENTE NON AVVIENE...

L'Edicola



Una nuova rivista per Amiga edita, pensate un po', dalla ANTIC's, la casa produttrice di Zoetrope, è uscita sul mercato statunitense e canadese.

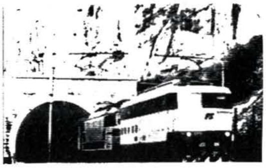
Anche in Italia qualche fortunato potrà trovarne una copia dal giornalaio, distribuita dalla Interdipress.

Si tratta di un prodotto di buon livello tecnico, come la maggior parte delle riviste americane ed è corredata da un dischetto che può essere ordinato per posta.

Il numero 2, di cui potete vedere la copertina, è dedicato in parte alla grafica, ma non manca qualche recensione dettagliata di giochi, come FALCON, e novità hardware.

PAL-RGB CONVERTER

Con questo simpatico ed indispensabile apparecchietto potremo finalmente digitalizzare immagini a colori di qualità eccellente senza far ricorso a filtri colorati; sarà possibile vedere sia l'immagine proveniente dall'Amiga che quella proveniente dalla telecamera, semplicemente spostando un deviatoree. Potremo, inoltre, regolare il contrasto e la saturazione per ottenere sempre il risultato migliore. Ecco una delle immagini digitalizzate in fase di dimostrazione.





PERSONAL FONT MAKER

Il nuovo programma della Cloanto permette la creazione di set di caratteri personalizzati, l'ottimizzazione di un qualunque font esistente e la stampa nella migliore risoluzione di cui sia capace la stampante in uso.

E' praticamente indispensabile per il Desktop Publishing. Osservate voi stessi la prova di stampa eseguita con una pre-release del programma dallo stesso autore Michele Console Battilana con la stampante a 24 aghi Commodore 1224.

Maker PROVA DI STAMPA Cloanto
Personal Fonts Maker PROVA DI

Il punto sulla i

Readme N.0 Settembre 89:
Un bus a 16 bit non indirizza direttamente un Mb di memoria, ma 64K.

Readme N.1 Ottobre 89:
la velocità del microprocessore è determinata dal clock: in particolare quella del microprocessore 68030 inserito nelle shede di espansoione per A2000 della GVP è 25 megahertz.

PAG. 36 - il listato corretto della funzione Quadrato() è:

```
Quadrato(x)
int x;
{
return(x*x);
}
```

PAG. 64 - ColdReset: al secondo rigo Xdef e non Xref.

Questo numero di Readme è stampato su carta riciclata: abbiamo in questo modo voluto dare il nostro, seppur piccolo, contributo alla salvaguardia del patrimonio boschivo. Che ne pensate? Scriveteci.